Chapter 6
Graphics Native Approach to Identifying Surface Atoms of Macromolecules

Huagen Wan, Yunqing Guan, and Yiyu Cai

Abstract Classification of "surface atoms" or "interior atoms" of proteins or other macromolecules is significant for many biochemical tasks, particularly for molecular docking. We present a simple and easy-to-implement algorithm for identifying surface atoms of macromolecules from interior atoms. Unlike existing methods that are based on geometry computations, our approach takes the advantage of graphics hardware, and most of the computations are fulfilled with graphics processing unit (GPU). The algorithm can be easily incorporated within visualization applications for macromolecules to enable the removal of interior atoms from a macromolecular structure, thus simplifying the graphics display and manipulation.

Keywords Molecular surface • Solvent accessible surface • Surface atoms • Interior atoms • Graphics algorithm • Graphics hardware • Rendering

6.1 Introduction

The structure of proteins and other macromolecules is fundamental for the underlying biological interactions. As biological molecules interact at their surfaces, an understanding of the surface characteristics of the participating molecules would be particularly useful for studying interactions among them. Although the boundary surface of the electronic density surrounding a molecule is not well defined, the term of molecular surfaces was first introduced by Richards in 1977 to describe a molecular envelope accessible, e.g., by a solvent molecule [1]. There are several

H. Wan
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China 310027
e-mail: hgwan@cad.zju.edu.cn

Y. Guan
Institute for Media Innovation, Nanyang Technological University, Singapore, Singapore 637553
e-mail: yunqing.guan@ntu.edu.sg

Y. Cai (✉)
School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, Singapore 639798
e-mail: myycai@ntu.edu.sg

Y. Cai, S. See (eds.), *GPU Computing and Applications*,
DOI 10.1007/978-981-287-134-3_6

representational schemes to define the molecular surface model. These include the isovalue electronic density surface, van der Waals surface, Richards's molecular surface, and solvent accessible surface (SAS) [2].

The isovalue electronic density surface is described as the molecular envelope consisting points with the same electronic density values, generally 0.002 au, in a given volume.

The van der Waals surface is, however, defined as the molecular envelope containing the atomic spheres with van der Waals radii. It is simply constructed from overlapping van der Waals spheres of the atoms. Given the spherical representation of the atoms with van der Waals radii, the van der Waals surface is represented as the union of all portions of all atomic sphere surfaces not occluded by neighboring atomic spheres.

Richards's molecular surface is composed of two different kinds of surface patches: the contact surface and the reentrant surface [1]. Imagine the approach of a small "probe" molecule up to the van der Waals surface of a macromolecule. Depending on the size of the probe molecule (except for a probe of zero size), there will be regions of "dead space," crevices that are not accessible to the probe as it rolls about on the macromolecule. The molecular surface is traced out by the inward-facing part of the probe molecule sphere as it rolls on the van der Waals surface of the macromolecule. The contact surface is formed by the part of the van der Waals surface of each atom that is accessible to the probe sphere. The reentrant surface corresponds to the inward-facing part of the probe sphere when it is simultaneously in contact with two or three atoms forming crevices too narrow for the probe molecule to penetrate. Richards's molecular surface is usually defined using a water molecule as the probe, represented as a sphere with radius 1.4 Å. In [3], Connolly has proposed an analytical method for calculating Richards's molecular surface, with which a set of curved regions of spheres and tori, joined together at circular arcs, are used to describe the molecular surface.

The solvent accessible surface (SAS) corresponds to the molecular envelope of the surface that is traced by the center of the probe molecule sphere as it rolls on the van der Waals surface of the macromolecule [4, 5]. The center of the probe molecule can thus be placed at any point on the accessible surface and not penetrate the van der Waals spheres of any of the atoms in the macromolecule. Mathematically, it is equivalent to a van der Waals surface in which the atomic radii have been extended by the probe radius.

Figure 6.1 illustrates the last three kinds of representational schemes for the molecular surface model.

Molecular surface modeling has several applications. One direct benefit with molecular surfaces is the protein or macromolecule visualization [6–8]. Various physical chemical properties such as electrostatic potential and hydrophobicity [9] can be mapped onto the molecular surface and color coded [10–14]. Crucial in protein-protein interaction and interface study [15], molecular surfaces have been applied to the protein-protein docking problem which is the prediction of a complex between two proteins given the three-dimensional structures of the individual

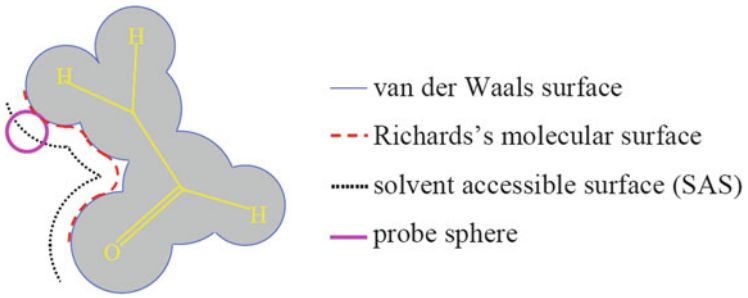

Fig. 6.1 Schematic view of van der Waals surface, Richards's surface, and SAS

proteins [16–18]. Identifying binding pockets on protein surfaces to help in rational or structure-based drug design [19–25] is another major purpose of molecular surface investigation.

For those atoms of a protein or other macromolecules, a significant number of them lie buried beneath the molecular surface of the protein or macromolecule. Interactions among these macromolecules are often dominated by interactions with the "surface atoms," although interactions with the interior atoms of the macromolecule certainly contribute to the total intermolecular interaction energy. Therefore, a classification with "surface atoms" or "interior atoms" of proteins or other macromolecules is significant for biochemical tasks, particularly for molecular docking. For such a classification, several factors should be considered, e.g., the running time of the classification algorithm, number of surface atoms correctly identified, and the numbers of surface atoms and interior atoms incorrectly identified [26].

In this chapter, we present a simple, graphics hardware-based approach to identifying surface atoms of macromolecules from interior atoms. The chapter is organized as follows. In Sect. 6.2, we review the related research works. In Sect. 6.3, we describe the overview of our algorithm as well as its implementation details. Section 6.4 presents some experimental results and discussions and the final section concludes our study.

6.2 Prior Work

Deanda et al. [26] propose a definition for surface atoms as follows: "An atom will be classified as an 'effective surface atom' if its SAS area is greater than a user specified minimum threshold value for the atomic SAS area $SA_{acc}^{\min}$." Accordingly, they develop an SAS approach to distinguishing the surface atoms of macromolecules from the interior atoms. The SAS approach is a computational one that calculates the atomic contributions to the SAS area and designating beforehand a constant value as the minimum threshold for the atomic SAS area. They adopt a

surface area and volume package (SAVOL3) [27, 28] to calculate the atomic SAS area. In their paper, they also summarize several other methods for surface atom identification: (1) the NIN (number of intersecting neighbors) approach based on the intuitive notion that the number of intersecting neighbors (i.e., atomic spheres intersect one another) would be far greater for interior atoms than for surface atoms, (2) the SOV (sum of vectors) approach which is a variation of the NIN approach and uses the norm of the SOV to its neighbors as a criterion for classifying surface atoms from interior atoms, (3) the UCSF (University of California at San Francisco) approach that imbeds the macromolecule within a 3D lattice and associates the atoms with the lattice points for classifying surface atoms [29], and (4) the MDS (molecular dot surface) approach which uses the molecular cloud point representation to identify surface atoms [30].

All those algorithms are geometry based. While the NIN, SOV, UCSF, and MDS approaches suffer from ambiguities for identifying surface atoms (i.e., atoms are often misclassified) [26], the SAS approach needs geometry computations of atomic SAS areas which are often performed with specific software packages. With the rapid development of graphics processing unit (GPU), numerous applications have been developed based on graphics hardware [31–35, 39–42]. We believe that techniques developed for graphics hardware rendering will be very useful for bio-related tasks, such as the identification of surface atoms for proteins or other macromolecules.

6.3 Algorithm Overview and Implementation

The kernel idea behind the definition of surface atoms in [26] is that if an atom of a macromolecule contributes to the molecule's SAS, then the atom will be considered as a "surface atom" of the molecule. Bearing this in mind, we adjust slightly the surface atom definition as follows. Let an atom A (with van der Waals radius r) of a macromolecule M be represented as a hard sphere HS and the counterpart of HS with the radius being extended by the probe radius pr to $(r + pr)$ be denoted as an extended hard sphere (EHS), and then atom A will be classified as a "surface atom" if EHS can be seen from outside of the solvent accessible surface (SAS) of the molecule M.

6.3.1 Algorithm Overview

Our algorithm is based on the rendering of the EHSs with commercially available graphics hardware. Therefore, we can exploit the hardware to increase performance.

Imagine that the solvent accessible surface of a macromolecule M is surrounded by a bounding box and that each face of the box is a viewing plane. An image is

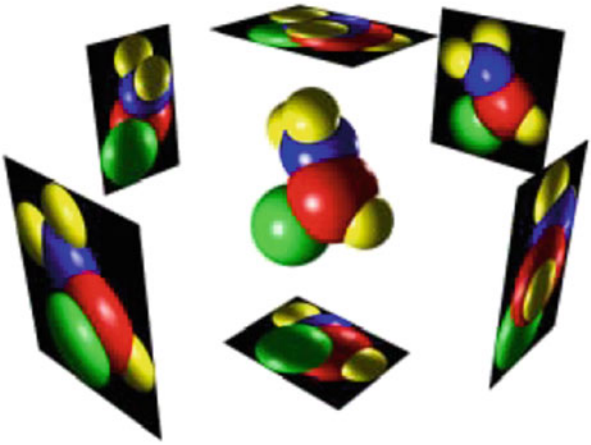

Fig. 6.2 Identifying surface atoms with color and depth buffers

generated for each face by parallel projecting onto it the *EHS*s of the macromolecule M with hidden surfaces removed by depth comparison (Fig. 6.2).

Therefore, if the *EHS* of an atom appears in one or more of the six images, then the atom will be classified as a surface atom. Resolutions for the faces are chosen so that there are enough pixels for classifying the surface atoms.

6.3.2 Implementation

The implementation of the algorithm takes the advantage of graphics hardware capabilities (e.g., color buffer and depth buffer), OpenGL graphics library as well as the OpenGL utility toolkit (GLUT) [36, 37]. Apart from the objects positioning and orientation in the scene, OpenGL offers facilities to define a viewing volume and to specify the way objects are projected on the screen. There are two kinds of projection: orthographic and perspective. The orthographic projection draws object without affecting their relative size. The perspective projection is similar to our vision mode: the further an object is, the smaller it appears, and two parallel straight lines seem to converge in the distance. In both cases, viewing volumes are hexahedra: a box or a truncated pyramid respectively (Fig. 6.3).

In our algorithm, the orthographic projection is used and the bounding box of the macromolecule's SAS is adopted as the viewing volume. An image is generated for each of the six faces of the viewing volume by rendering the *EHS*s of the macromolecule with hidden surfaces removed.

For graphics hardware rendering with OpenGL, the color information at each pixel can be stored either in RGBA mode or in color-index mode. In the first mode, the R, G, B, and possibly alpha values are kept for each pixel. In the second mode, however, only a single number (called the color index) is stored for each pixel. Each color index indicates an entry in a color table that defines a particular set of R, G,

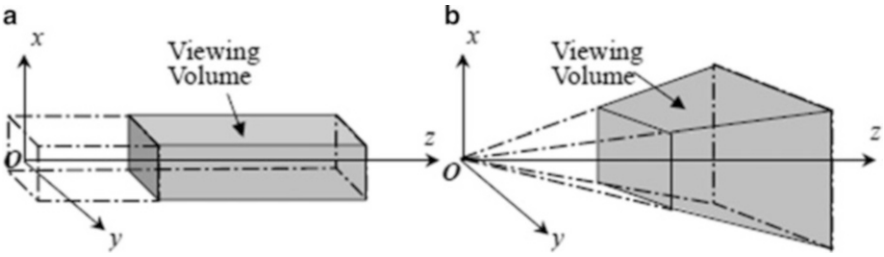

Fig. 6.3 (**a**) Orthographic and (**b**) perspective views

and B values. In either RGBA or color-index mode, a certain amount of color data is stored at each pixel. This amount is determined by the number of bitplanes in the frame buffer. A bitplane contains one bit of data for each pixel.

For most commonly available low-end graphics cards, at least 16 bitplanes are provided for color storage in RGBA mode, and at most 8 bitplanes are available in color-index mode. Considering there are often several hundreds to thousands of atoms in a typical macromolecule, we choose the RGBA mode in this implementation. It would be more straightforward with the color-index implementation, and high-end graphics workstations can be used to improve its efficiency (e.g., with 12 bitplanes on SGI Octane workstations for color-index buffers).

Each atom is firstly initialized with a unique identity, and a color table (with the number of atoms of the macromolecule in size) is created with each of its components corresponding to an atom identity, and then each atom's *EHS* is rendered with the color (in the color table) corresponding to the atom's identity. Subsequently, the color values of the rendered atoms' *EHS*s are read from the color buffer and used to determine the appearance of the *EHS*s in the images. To do so, a Boolean array is used as a flag list to indicate which atom is a surface atom and which one is not. The display list is used for rendering *EHS*s with a high performance.

It is worth noting that the same viewing matrix is used for a pair of rendering (e.g., front and back, left and right, and top and bottom). This is done by setting the depth comparison logic on one of the renderings to save the z-depth values farthest away instead of closest with glDepthFunc( ) and set the face culling logic on the same rendering to eliminate the front polygons of *EHS*s with glCullFace( ). For instance, when rendering the two images for the front and back pair of the viewing volume, firstly the viewing matrix for the front view is set, and then the first image (corresponding with the front view) is generated by culling back polygons (of *EHS*s of the molecule) which face away from the front view and setting the depth comparison logic to GL_LEQUAL to make the depth test satisfied if the incoming z value is less than or equal to the stored z value and finally the second image (corresponding to the back view) is rendered by culling front polygons which face toward the front view and setting the depth comparison logic to GL_GREATER to make the depth test passed if the incoming z value is greater than the stored z value.

Figure 6.4 lists the pseudo code of our algorithm.

```
01  proc IDENTIFYING_SURFACE_ATOMS()
02      Initialize macromolecule data;
         /* e.g. read data file, calculate the bounding box of
            the molecule's SAS, assign an identity for each of the
            atom and build the color table, and etc. */
03      Initialize graphics;
         /* e.g. set color display mode, open display window,
            enable depth test and culling test,
            compile display list for EHS, and etc. */
04      Initialize a flag array for identified surface atoms;
         /* each value of the array is initialized to false to
            indicate that all the atoms are not surface atoms */
05      for each pair of the box faces
            /*e.g. front & back, left & right, and top & bottom */
06          Set viewing matrix with the bounding box parameters;
07          for each face of the pair
08              Set the rendering attributes;
                 /* e.g. face culling logic,
                    depth comparison logic, and etc. */
09              Clear color buffer & depth buffer;
10              for each EHS of the macromolecule
11                 Set rendering color to the EHS's given color;
12                 Render the EHS;
13              end_for
14              Read color values from the color buffer;
15              Update the flag array with the color values read;
                  /* if the EHS of an atom appears in the image,
                     change the value of the array element
                     corresponding to the atom to true */
16          end_for
17      end_for
18      Output the result;
19  end_proc
```

Fig. 6.4 Pseudo code of our algorithm

6.3.3 Improvements

The above algorithm can quickly and successfully classify most surface atoms of any macromolecules. The main limitation of the above approach is that it may miss concavities. If some *EHS* of an atom contributes to the molecule's SAS and is not visible from any of the six faces of the viewing volume, then this atom will not be properly classified. The algorithm, however, can be easily improved by adding more viewing planes. For instance, we can sample from the four diagonals of the above bounding box to add 8 more viewing directions and construct viewing planes to render the atoms' *EHS*s (Fig. 6.5). Furthermore, we find using higher resolution of the viewing plane can also improve the classification. We will show with experiment how they help in the next section.

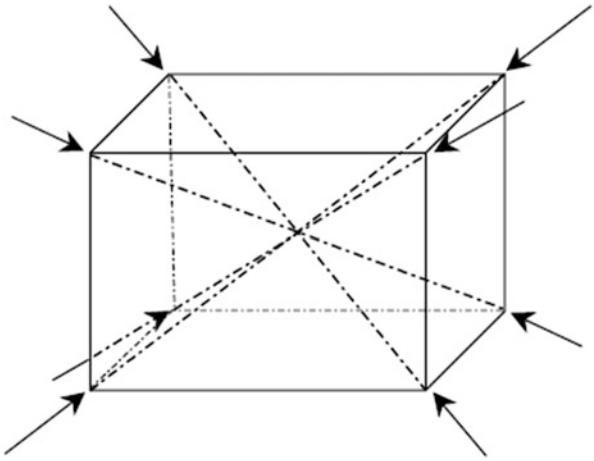

Fig. 6.5 Improving the algorithm by sampling from additional 8 viewing directions

6.4 Experimental Results and Discussions

Several macromolecules from the Protein Data Bank (PDB) [38] were tested under the resolution of 1792 * 1344 in the true color mode (32-bit mode). Figure 6.6 shows the contents of the color buffer when performing the test with a triose-phosphate isomerase (1TIM). Table 6.1 lists the testing results with a dihydrofolate reductase (1RA2), a thermolysin (7TLN), and a triose-phosphate isomerase (1TIM). The tests were performed under different resolutions of the viewing plane (e.g., 100 * 100, 400 * 400, 800 * 800, 1000 * 1000, and 1182 * 1182) and with different configurations of viewing planes (e.g., 6 viewing planes and 14 viewing planes). For comparison reason, the experimental data of the SAS approach selected from [26] were listed in Table 6.2. Their experiments were performed on an SGI Indigo with an R4400 processor.

From Table 6.1, we can clearly see that the number of classified surface atoms increases with the increment of both the viewing planes and the rendering resolution. However, while the accuracy of the classification is nearly constantly improved with more sampling view planes, the number of classified surface atoms increases nonlinearly with the increment of the rendering resolution. For the number of classified surface atoms of the 3 testing macromolecules, there is only a subtle degree of difference for the resolutions of 1000 * 1000 and 1182 * 1182.

Theoretically, there may be an “accurate” or “exact” number of surface atoms for a macromolecular structure, and there may exist a “clear” borderline between surface atoms and interior atoms. However, to our knowledge, there is yet to have a theoretical solution at present time to calculate the “accurate” or “exact” surface atom number. It is a challenging job as well to numerically find out this “accurate” or “exact” number and/or “clear” borderline. In fact, the accuracy of the SAS approach [26] is dependent upon the user-specified minimum threshold value for

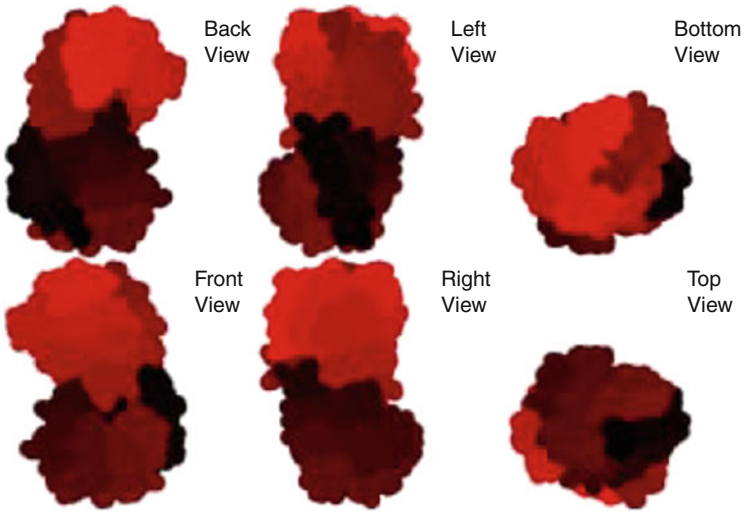

Fig. 6.6 Color buffer contents when testing with triose-phosphate isomerase (1TIM)

the atomic SAS area and the precision of the atomic SAS area calculation. On the other hand, the accuracy of our graphics hardware-based approach depends upon both the viewing plane setting and the rendering resolution. Still, we think that the numerical solutions are worth trying when "accurate" theoretical solutions are not available. Also, we believe that the numbers of classified surface atoms from our approach show kinds of tendency of convergence when the viewing directions and resolution are increased. This again turns out as an interesting yet difficult research topic.

Conclusions

This chapter presents a fast and easy-to-implement algorithm for identifying surface atoms of macromolecules from interior atoms, which is based on the color buffer and z-buffer. The algorithm can be easily incorporated within visualization applications for macromolecules as a preprocessing step to enable the removal of interior atoms from the macromolecular structure. Doing so, a simplified macromolecular structure can be generated for graphics display which can reduce the time required for display and manipulation of macromolecules.

Unlike existing methods for identifying surface atoms of macromolecules mainly based on geometry computations performed by general CPU, our approach takes the advantage of widely available graphics hardware and most of the computations are fulfilled with the graphics processing unit (GPU). As our algorithm is based on the color buffer and z-buffer, its

(continued)

Table 6.1 Experimental results with several macromolecules from Protein Data Bank

	Resolution	100 × 100		400 × 400		800 × 800		1000 × 100		1182 × 1182	
Molecule	Number of sampling viewing planes	6	14	6	14	6	14	6	14	6	14
IRA2 Total atom number: 1268	Number of surface atoms	682	703	723	752	732	756	732	762	733	763
	Running time in seconds	0.34	0.81	0.48	1.14	0.91	2.20	1.34	3.08	1.83	4.20
7TLN Total atom number: 2436	Number of surface atoms	1006	1049	1107	1155	1120	1172	1125	1174	1128	1175
	Running time in seconds	0.63	1.48	0.77	1.83	1.16	2.89	1.45	3.69	1.92	4.66
ITIM Total atom number: 3740	Number of surface atoms	1463	1594	1680	1861	1721	1829	1730	1844	1730	1852
	Running time in seconds	0.98	2.25	1.08	2.63	1.39	3.69	1.64	4.48	1.92	5.36

Table 6.2 Experimental data of the SAS approach (selected from [26])

Molecule	Threshold	$SA_{\min}^{acc} = 1\text{A}^2$	$SA_{\min}^{acc} = 0.01\text{A}^2$
IRA2 Total atom number: 1268	Number of surface atoms	604	730
	Running time in seconds	7.57	7.57
7TLN Total atom number: 2436	Number of time in seconds	891	1181
	Running time in seconds	15.73	15.73
ITIM Total atom number: 3740	Number of surface atoms	1565	2216
	Running time in seconds	23.45	23.45

complexity is independent of the molecule complexity but dependent on the rendering resolution and its viewing plane setting.

With the computational power of graphics hardware outperforming that of general CPU by Moore's law [34], we believe that algorithms based on GPU for biochemical tasks will be very promising in the future.

Acknowledgments The authors would like to thank the partial funding support from Singapore MOE Tier 1 (RG 10/12).

References

1. Richards, F.M.: Areas, volumes, packing, and protein structure. Ann. Rev. Biophys. And Bioeng. **6**, 151–176 (1977)
2. Leach, A.R.: Molecular Modelling: Principles and Applications, 2nd edn. Essex, Pearson Education EMA (2001)
3. Connolly, M.L.: Analytical molecular surface calculation. J. Appl. Crystallogr. **16**, 548–558 (1983)
4. Lee, B., Richards, F.M.: Interpretation of protein structures: estimation of static accessibility. J. Mol. Biol. **55**, 379–400 (1971)
5. Hermann, R.B.: Theory of hydrophobic bonding. II. The correlation of hydrocarbon solubility in water with solvent cavity surface area. J. Phys. Chem. **76**, 2754–2759 (1972)
6. Quarendon, P.: A general approach to surface modeling applied to molecular graphics. J. Mol. Graph. **2**, 91–95 (1984)
7. Connolly, M.L.: Depth buffer algorithms for molecular modeling. J. Mol. Graph. **3**, 19–24 (1985)
8. Connolly, M.L.: Plotting protein surfaces. J. Mol. Graph. **4**, 93–96 (1986)
9. Ooi, T., Oobatake, M., Nemethy, G., Scheraga, H.A.: Accessible surface areas as measure of the thermodynamic parameters of hydration of peptides. Proc. Natl. Acad. Sci. USA **84**, 3086–3090 (1987)
10. Chapman, M.S.: Mapping the surface properties of macromolecules. Protein Sci. **2**, 459–469 (1993)
11. Nicholls, A., Bharadwaj, R., Honi, B.: GRASP: graphical representation and analysis of surface properties. Biophy. J. **64**, A166 (1993)

12. Heiden, W., Moeckel, G., Brickmann, J.: A new approach to analysis and display of local lipophilicity/hydrophilicity mapped on molecular surfaces. J. Comput. Aided Mol. Des. **7**, 503–514 (1993)
13. Duncan, B.S., Macke, T.J., Olso, A.J.: Biomolecular visualization using AVS. J. Mol. Graph. **13**(5), 271–282 (1995)
14. Altman, R.B., Hughes, C., Gerstein, M.B.: Methods for displaying macromolecular structural uncertainty: application to the globins. J. Mol. Graph. **13**, 142–152 (1995)
15. Janin, J., Chothi, C.: Surface, subunit interfaces and interior of oligomeric proteins. J. Mol. Biol. **204**, 155–164 (1988)
16. Zielenkiewicz, P., Rabczenko, A.: Protein-protein recognition: method for finding complementary surfaces of interacting proteins. J. Theor. Biol. **111**, 17–30 (1984)
17. Santavy, M., Kypr, J.: A fast computer algorithm for finding an optimum geometrical interaction of two macromolecules. J. Mol. Graph. **2**, 47–49 (1984)
18. Cherfils, J., Janin, J.: Protein docking algorithms: simulating molecular recognition. Current Opinion in Structural Biology **3**, 265–269 (1993)
19. Kuntz, I.D.: Structure-based strategies for drug design and discovery. Science **257**, 1078–1082 (1992)
20. Navia, M.A., Murcko, M.A.: Use of structural information in drug design. Curr. Opin. Struct. Biol. **2**(2), 202–210 (1992)
21. Bugg, C.E., Carson, W.M., Montgomery, J.A.: Drugs by Design. Scientific American **269**(6), 92–98 (1993)
22. Murcko, M.A., Rotstein, S.H.: GenStar: a program for de novo drug design. J. Comput. Aided Mol. Des. **7**, 23–43 (1993)
23. Verlinde, C.L.M.J., Hol, W.G.J.: Structure-based drug design: progress, results and challenges. Structure **2**, 577–587 (1994)
24. Whittle, P.J., Blundell, T.L.: Protein structure-based drug design. Ann. Rev. Biophys. Biomol. Struct. **23**, 349–375 (1994)
25. Jackson, R.M., Sternberg, M.J.E.: Protein surface-area defined. Nature **366**(6456), 638 (1993)
26. Deanda, F., Pearlman, R.S.: A novel approach for identifying the surface atoms of macromolecules. J. Mol. Graph. Model. **20**, 415–425 (2002)
27. Pearlman, R.S.: Molecular surface area and volume: their calculation and use in predicting solubilities and free energies of desolvation. In: Dunn III, W.J., Block, J.H., Pearlman, R.S. (eds.) Partition Coefficient: Determination and Estimation, pp. 3–20. Pergamon Press, New York, NY (1986)
28. Savol3: surface & volume algorithms, http://www.chem.ac.ru/Chemistry/Soft/SAVOL3.en.html. Last visit 4 Oct 2003
29. Bash, P.A., Pattabiraman, N., Huang, C., Ferrin, T.E., Langridge, R.: van der Waals surfaces in molecular modeling: implementation with real-time computer-graphics. Science **222**, 1325–1327 (1983)
30. Brusniak, M.-Y.K.: Development and application of software for CADD. Ph.D. Dissertation, The University of Texas, Austin (Chapter 2) (1996)
31. Baxter, W.V., III, Sud, A., Govindaraju, N.K., Manocha, D.: GigaWalk: interactive walkthrough of complex environment. UNC-CH Technical Report TR02-013 (2002)
32. Govindaraju, N.K., Sud, A., Yoon, S.E., Manocha, D.: Parallel occlusion culling for interactive walkthroughs using multiple GPUs. UNC Computer Science Technical Report TR02-027 (2002)
33. Karabassi, E.A., Papaioannou, G., Theoharis, T.: A fast depth-buffer-based voxelization algorithm. J. Graph. Tools **4**(4), 5–10 (1999)
34. Lin, M., Manocha, D.: Interactive geometric computations using graphics hardware. In: Siggraph'2002 course notes (2002)
35. Tomas, A.M., Eric, H.: Real-Time Rendering, 2nd edn. A.K. Peters Ltd, Natick, MA (2002)
36. McReynolds, T.: Programming with OpenGL: Advanced Rendering. In: SIGGRAPH'96 course notes (1996)

37. GLUT specification, http://www.opengl.org/developers/documentation/glut/index.html. Last visit 4 Oct 2003
38. Protein Data Bank, http://www.rcsb.org/pdb/. Last visit 4 Oct 2003
39. Colberg, P. H., Höfling, F.: Highly accelerated simulations of glassy dynamics using GPUs: Caveats on limited floating-point precision. Comp. Phys. Comm. **182** (5), 1120–1129, (2011)
40. Ufimtsev, I.S., Martinez, T.J.: Graphical processing units for quantum chemistry. Comp. Sci. Eng. **10**(6), 26–34 (2011)
41. Pronk, S., Larsson, P., Pouya, I., Bowman, G.R., Haque, I.S., Beauchamp, K., Hess, B., Pande, V.S., Kasson, P.M., Lindahl, E.: Copernicus: a new paradigm for parallel adaptive molecular dynamics. In: 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–10, 12–18 (2011)
42. Dror, R.O., Dirks, R.M., Grossman, J.P., Xu, H., Shaw, D.E.: Biomolecular simulation: a computational microscope for molecular biology. Annu. Rev. Biophys. **41**, 429–452 (2012)