

Measuring Properties of Molecular Surfaces Using Ray Casting

Mike Phillips^{*†}, Iliyan Georgiev[†], Anna Katharina Dehof[‡], Stefan Nickels^{‡§},
Lukas Marsalek[†], Hans-Peter Lenhof[‡], Andreas Hildebrandt[‡], Philipp Slusallek^{*†§}

**DFKI Saarbrücken*

†Computer Graphics Lab, Saarland University

‡Center for Bioinformatics, Saarland University

*§Intel Visual Computing Institute, Saarland University
66123 Saarbrücken, Germany*

Abstract—Molecular geometric properties, such as volume, exposed surface area, and occurrence of internal cavities, are important inputs to many applications in molecular modeling. In this work we describe a very general and highly efficient approach for the accurate computation of such properties, which is applicable to arbitrary molecular surface models. The technique relies on a high performance ray casting framework that can be easily adapted to the computation of further quantities of interest at interactive speed, even for huge models.

Keywords—ray tracing; surface area; volume;

I. INTRODUCTION

Understanding and predicting the interactions between biomolecules is one of the most challenging – and the most important – tasks science is faced with today. To a large extent these interactions are determined by the three-dimensional structures of the component molecules and, consequently, the efficient and accurate computation of molecular geometric properties has long since been studied extensively in Bio- and Cheminformatics.

One typical use-case for simple geometric properties is the estimation of binding free energies in the presence of a solvent – water in the biomolecular case. To occur in a certain shape, each biomolecule has to displace a number of water molecules to form a cavity for itself. This loss of freedom of the solvent leads to an entropic effect that is a function of the molecular volume. Similarly, the surface tension of the water surrounding two individual molecules will differ from the tension around their complex, leading to a term in the free energy of binding that is a function of the solvent-exposed surface area.

Other important geometric properties often focus on the presence, volume, and surface area of possible internal cavities, pockets, or tunnels inside the molecule of interest.

The computation of such properties requires a definition for the molecular shape, i.e. at least a description of its surface. On the other hand, the concept of a 'molecular surface' is far from being uniquely defined: with atoms being quantum mechanical objects that have no well-defined boundaries, and with molecules being highly flexible arrangements

of such atoms for which a single geometric arrangement will always be an approximation, any sharp molecular boundary can only be an approximate model for the overall shape of the molecule. This has led to the definition of a number of important shape models for molecules (c.f. Section II), each with its own advantages and disadvantages, and different areas of application. However, previous work on volume, area, and cavity detection in molecular scenarios has usually focused on select models, restricting their applicability. In addition, with each new surface definition, new techniques for area and volume computation as well as cavity detection have to be developed. A notable example are grid-based techniques, which allow, in principle, to estimate the desired quantities on arbitrary geometries, albeit with relatively low precision or high computational effort.

In this work we propose to use ray casting techniques, as known from computer graphics, in order to accurately estimate a variety of geometric properties for arbitrary molecular surface definitions. Ray casting methods are known to parallelize very well and are able – using suitable acceleration structures – to handle even huge geometric models at interactive speeds. The only requirement posed on the molecular representation is the ability to efficiently intersect it with a ray of arbitrary direction, which is trivially possible for all tessellated surfaces, but also for more general representations. Ray casting of molecular surfaces has previously been used for visualization [1], [2], but, to the best of our knowledge, not for computational purposes.

In the following, we will first present some of the most important previous work on molecular volume, area, and cavity computation. We will then discuss our new ray casting method for this task before we demonstrate our results on a number of triangulated protein surfaces.

II. BACKGROUND AND RELATED WORK

In the past, many approaches for surface area and volume estimation of molecules have been proposed. To understand these previous approaches, it is important to note that the nomenclature in the field is highly confusing, in particular in earlier works. The reason for the ambiguities in notation

comes from different definitions for the molecular surface of interest. The only surface that is uniquely defined in all publications is the *van-der-Waals surface* (vdW) (c.f., for instance, [3]), i.e. the surface of the union of all atomic spheres. Other surfaces are derived from the solvent’s point of view: the *solvent accessible surface* (SAS) is defined as the trace of the center of a sphere representing a solvent molecule that is rolled over the van-der-Waals surface of the molecule of interest [4]. In contrast the *solvent excluded surface* (SES) is defined as the trace of the inward-facing surface of the probe sphere [5], [6], [7], [8] in the same process. More recent surface definitions are skin surfaces [9], [10], alpha shape [11], beta shape [12], and minimal molecular surfaces [13]. Here, it is important to keep in mind that no single surface definition is ‘correct’ or ‘incorrect’ in the strict sense of the meaning - not only are molecules dynamic rather than static, but also the simple picture of atoms as spheres with a given position and radius is a crude approximation of quantum mechanics. Hence, all of the above definitions may be more or less appropriate to a given task at hand. However, even though more modern surface definitions usually possess improved smoothness properties or lead to better triangulations, they seem much less popular in real-world applications than the traditional vdW, SAS, and SES surfaces which have an intuitive biological interpretability.

The diversity of surface definitions, and the widely varying quality of surface point distributions or triangulations, has led to the development of a broad range of methods for molecular surface computation, e.g., based on overlapping spheres [14], the Gauss-Bonnet theorem [15], [16], space transformation [17], contour-buildup [18], alpha shape theory [11], or variable probe radii [19]. These different methods yield point clouds, triangulations, or polyhedral meshes as a representation of the molecular surfaces or volumes [20], [7], [8], [21], [22], or use grid-based constructions [23], or analytical descriptions [24], [8].

For a particular surface definition, the molecular volume is defined as the volume enclosed by its surface, inducing again a number of different volume definitions, e.g., the van-der-Waals volume is defined as the volume of the union of overlapping van-der-Waals spheres [24]. The SAS (SES) volume is defined as the volume within the SAS (SES) of a molecule [25].

For computing molecular volumes, two general approaches may be distinguished: numerical and analytical. The analytical algorithms describe the volume by a set of equations [7], [25], [24] whereas the numerical algorithms approximate the volume by smaller geometric objects [26], [27].

One of the oldest is the Voronoi polyhedral volume computation VOLUME of Richards [26] that is for example used by the Voss Volume Voxelator and the Vadar [28] server. It divides the space between atoms by bisector planes

into polyhedra. Using those polyhedra the volume can be computed.

A very elegant approach for the computation of solvent excluded and solvent accessible surface is based on the so-called reduced surface as implemented in the MSMS program of Michael Sanner [8]. Despite its name, the reduced surface is actually a graph structure, where nodes, edges, and faces represent regions where the solvent sphere touches one, two, or multiple surface atoms.

In addition to building surface triangulations, MSMS also estimates the volume and surface area of each component of the molecular surface, including internal cavities.

A. Ray Casting

Ray casting is a general method, mostly used in computer graphics, for determining point to point visibility in space [29]. The algorithm proceeds by taking a ray and finding the first intersection with an object along the direction of the ray. This is accomplished by explicitly performing intersection tests between the ray and the geometry defined in the scene. The intersection test must be individually specified for each type of geometry used. The most common type of geometry is the triangle mesh, but intersection tests have been developed for other types, such as quadrics, implicit and parametric surfaces. See [29] for further details.

The recent developments of highly optimized algorithms and the advances in hardware have enabled ray casting to achieve real-time performance [30], [31] and to become a viable alternative to rasterization for interactive applications. A key intrinsic property of ray casting is its trivial parallelization, and optimized acceleration structures allow for sub-linear scalability with model size. In this paper, we apply ray casting to the problem of measuring properties of molecular surfaces.

III. ESTIMATING VOLUME INTEGRALS

A volume can be defined as the set, $\mathbb{S} \in \mathbb{R}^3$, of all points in space which lie within a region of interest, in this case, the molecular surface. The total volume of the set can be determined by

$$V = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y, z) dx dy dz, \quad (1)$$

where

$$f(x, y, z) = \begin{cases} 1 & \text{if } (x, y, z) \in \mathbb{S}, \\ 0 & \text{if } (x, y, z) \notin \mathbb{S} \end{cases} \quad (2)$$

is the characteristic function of \mathbb{S} . Using the fundamental theorem of calculus, we can calculate the volume of any enclosed region of space, provided that the characteristic function can be solved for any given point. Note that in one dimension (i.e. along a straight line) f is piece-wise constant. We reformulate Equation 1:

$$V = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) dx dy, \quad (3)$$

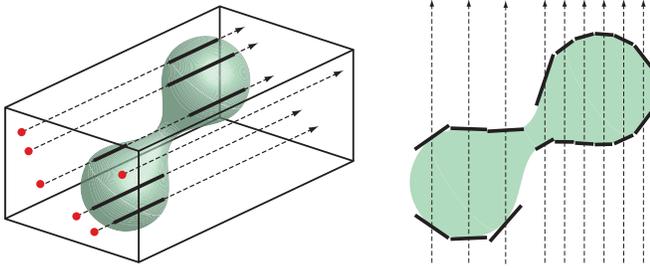


Figure 1. *Left*: Estimating the volume enclosed by a surface using ray casting. As rays intersect the surface, we track the total distance traveled inside the volume. *Right*: Approximating a surface with planar patches by back-projecting the ray footprints. Note how the patch size varies with the sampling frequency and surface orientation.

where

$$g(x, y) = \int_{-\infty}^{\infty} f(x, y, z) dz. \quad (4)$$

Here, $g(x, y)$ is a line integral for any line parallel to the Z-axis and passing through point (x, y) on the X-Y plane. In the following sections we describe how we solve Equation 3 and Equation 4 efficiently using ray casting.

A. The Line Integral

When working with an explicit representation of a 3D object (e.g. its surface triangulation), we do not have an explicit representation of its associated characteristic function f . In this section we show how to solve Equation 4 analytically using ray casting, without the need for explicitly defining f .

A ray in space is defined as

$$r(t) = \vec{o} + t\vec{d} \quad (5)$$

where \vec{o} is its origin, \vec{d} is its direction, and t is the ray parameter, or distance along the ray. If we define $r_{x,y}(t)$ to be a ray with origin $(x, y, 0)$, i.e. on the X-Y plane, and direction $(0, 0, 1)$, then Equation 4 can be rewritten as

$$g(x, y) = \int_{-\infty}^{\infty} f(r_{x,y}(t)) dt. \quad (6)$$

With the formulation of f in Equation 2, $g(x, y)$ gives the total distance that the ray traverses inside the volume.

The given surface representation defines the boundaries of the set \mathbb{S} , and thus the points in space where the characteristic function f changes. Considering that f is piecewise constant along a straight line, we can thus analytically integrate f for a given ray if we know the exact intervals where f is non-zero along the ray. To do this, we cast the ray and find all intersections with the surface. Whenever the ray intersects the surface, we track whether the ray enters or exists the enclosed volume, and accumulate the total distance inside the volume. The distance is calculated by simply subtracting each pair of exit and entry point values of the ray parameter t .

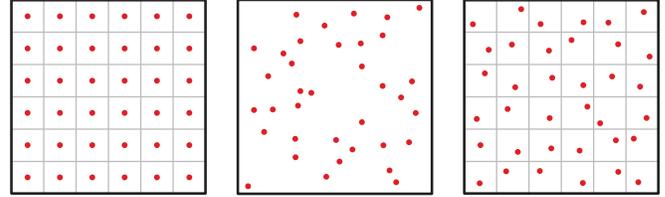


Figure 2. Different sampling methods. *Left*: Uniform sampling places samples on a regular grid inside the integration domain. *Middle*: Random sampling picks samples randomly within the domain. *Right*: Stratified random sampling combines the two approaches to achieve a statistically better sample distribution, resulting in a higher order of convergence.

B. Numerical Integration

Recall that the volume enclosed by a given surface is given by the integral in Equation 3. We estimate this integral by numerical integration, i.e. by evaluating $g(x, y)$ on N sample points and approximating the total volume by averaging the sampled function values in the following way:

$$\bar{V} = \frac{A}{N} \sum_{i=0}^N g(x_i, y_i) \approx \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) dx dy. \quad (7)$$

Here, A is the area of the side of the axis-aligned bounding box encompassing the surface that is parallel to the X-Y plane. In practice, we sample N points, (x_i, y_i) , on the side of the bounding box and shoot a ray to compute $g(x_i, y_i)$ as described in the previous section. Figure 1 left illustrates this process.

In Equation 7, the uniform scaling factor A/N can be thought of as the ray “footprint”.

Depending on the way of picking samples, \bar{V} can be viewed as a rectangle quadrature rule [32] (when sampling uniformly) or as a (quasi-) Monte Carlo estimator [33], [34] (when drawing samples from a (quasi-) random sequence). Figure 2 illustrates three different sampling strategies.

IV. EXTENSIONS

The method described in the previous section can be easily extended to also compute other properties, such as the surface area of the volume, as well as to find any cavities present within the volume.

A. Surface Area Estimation

If we assume that the surface is flat in the neighborhood of a ray intersection point, then we can locally approximate the surface with the tangent plane at that point. We can now estimate the surface area covered by the ray footprint, A/N , by back-projecting it onto the tangent plane. The total area, S , of the surface can then be approximated by

$$S \approx \bar{S} = \frac{A}{N} \sum_{i=0}^M \frac{1}{|\cos \theta_i|}, \quad (8)$$

where M is the total number of surface intersection points among all rays, and $\cos \theta_i$ is the cosine of the angle between

the corresponding ray direction and surface normal, which in our implementation is directly provided by the surface triangulation. With this process we effectively approximate the surface with M planar patches and sum up their areas. The more densely we distribute the rays, the more closely the approximation resembles the surface (see Figure 1 right).

Care must be taken where $\cos\theta_i$ approaches zero as the surface area estimate will approach infinity. We have found that clamping $\cos\theta_i$ to a small value (e.g. 0.07) eliminates the problem without introducing excessive error.

B. Cavity Detection

Cavity detection can be performed by populating a secondary data structure during ray casting, which can then be manipulated to reveal the cavities. We use a uniform 3D grid which discretizes the integration domain of the volume integral. Since our rays are parallel to the Z-axis, we can construct the grid such that there is exactly one ray per row in the Z-direction. Whenever a ray intersects the surface we update each cell the ray crossed between the previous and the current intersection points by writing the distance to the previous intersection point and a ray state. This state can be either inside the surface or in a potential cavity. After ray casting, in order to find the cavities using the grid data structure, we perform a 3D floodfill starting from a cell known to be outside the volume. All connected cells previously marked as a potential cavity will be re-marked as outside the volume by the floodfill algorithm. At the end, all cells that remain marked as potentially inside a cavity are guaranteed to be inside cavities.

We can enumerate the cavities by traversing the data structure. Each time a cavity cell is encountered the floodfill operation is once again performed, but this time the cells are re-marked as being part of a specific cavity.

Storing the distance values and cavity indices in the grid cells allows us to perform calculations on each cavity to determine its individual volume and surface area.

V. RESULTS

We have chosen several molecules of the dataset described in [35] for testing and comparing our results to. In order to ensure a fair comparison we used the program MSMS [8] to produce both reference estimates for volumes and areas, and the triangulated surfaces we test our methods with. Preprocessing was done in the following manner: we downloaded the molecules from PDB [36], checked the structure against BALL’s Fragment database, added missing hydrogens, and deleted ligand, cofactors, and water molecules using BALL [37], the biochemical algorithms library. We then ran MSMS to compute the surface area, volume, and triangulation of the SES surfaces with all contained cavities. We used the real-time ray tracer RTfact [38] to implement our ray casting based methods.

All testing was done on a computer with an Intel Core 2 Quad at 2.66GHz with 4GB of RAM, using a single core.

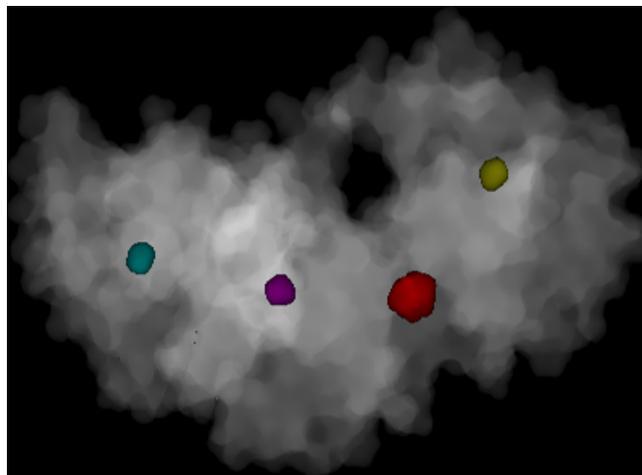


Figure 3. The volume of molecule 1g2a computed per pixel. Four cavities have been detected.

A. Volume Estimation

We have found that our volume estimates match very closely MSMS, even when using a low sampling density. The influence of different sampling resolutions upon the volume computation is summarized in Table IV and a comparison with MSMS for selected molecules is shown in Table I. We have found that uniform sampling produces more stable and accurate estimates than random sampling, as shown in Figure 4. Stratified or low-discrepancy sampling will likely further improve the accuracy of the estimates.

B. Surface Area Estimation

We have implemented the surface area estimation extension as a proof of concept to show that we can use our general ray casting framework to measure the surface area of any triangulated volume (protein and cavity). Tables I and III show that our method is capable of making reasonable estimates. The larger errors in the surface area estimates compared to the volume estimates are likely due to inconsistent surface triangulation as well as the cosine clamping (see Section IV-A), rather than a principal problem with the methodology. Hence, a more careful implementation that performs better triangulation and patch approximation, or that compensates for the cosine clamping, will lead to further improved accuracy.

C. Cavity Detection

Similar to the area estimation, our implementation of cavity detection is also a proof of concept in the sense that it is not yet optimized algorithmically and numerically. Nonetheless, we found that it works very reliably, given that the input triangulations of the cavity regions are correct and well formed. (While this usually holds for the outer surface, we found that cavity triangulations often lack in quality and contain overlapping regions.) The uniform grid data structure

Table I

ABSOLUTE VALUES AND RELATIVE DIFFERENCES FOR VOLUMES AND SURFACE AREAS (SA) OF TEN MOLECULES, COMPARING OUR METHOD WITH MSMS [8]. OUR METHOD USES A UNIFORM SAMPLING RESOLUTION OF 200×200 .

Name	Volume	Volume Diff (%)	SA	SA Diff (%)
1ajs	114468	0.181	29670	4.526
1b12	121083	0.060	39507	3.556
1rh0	121438	0.079	31728	4.679
1dys	89417	0.134	24310	4.179
1qpa	88423	0.195	26102	6.167
1qaz	48579	0.073	13730	4.664
1hf0	49202	0.376	18982	7.052
1hf8	37444	0.111	12118	3.732
1amu	140874	0.118	43084	4.395
1gar	52562	0.000	16338	3.045

Table II

NUMBER OF CAVITIES ALONG WITH VOLUME AND SURFACE AREA DIFFERENCES FOR SELECTED MOLECULES COMPARED TO MSMS [8].

Name	Cavities	Vol. Diff.(%)	SA Diff.(%)
1g2a	4	-1.01	-14.75
1qjp	4	2.40	-7.52
1e02	5	-4.67	-12.84
256l	3	0.05	0.04
2ihl	1	2.51	-1.16
1ton	4	-4.29	-14.70
1gar	3	2.94	-1.13

also allows us to calculate the volume and surface area of each cavity individually, or all together. Table II summarizes our cavity detection results for seven molecules and includes the differences between the volumes and surface areas we measured and those obtained from MSMS. Figure 3 shows an example molecule with four cavities.

D. Performance and Discussion

We have observed interactive performance for all calculations. Table V summarizes the timings for molecule 1a4u. Similar performance was observed for all tested molecules. Further optimizations to our software could greatly improve the execution times of these calculations, although the performance of our current implementation, including volume and surface area estimation, and cavity detection, is interactive. We also observed that a low sampling density is sufficient for accurate estimates, resulting in low computational requirements.

We have also measured the run times of MSMS for computing the volumes, surface areas and cavities of our test set of molecules. While both our method and MSMS run fast, a direct and fair performance comparison is unfortunately not possible, as MSMS cannot be controlled to perform these measurements separately. Moreover, as in all existing approaches we know of, in MSMS such measurements are tightly coupled with the surface triangulation process. Therefore, we believe that in this respect the advantage of our method is that it is completely independent of the surface triangulation method and even the surface representation in

Table III

MINIMUM, MAXIMUM AND AVERAGE DIFFERENCES BETWEEN SURFACE AREAS MEASURED BY OUR METHOD AND MSMS [8] AT DIFFERENT UNIFORM SAMPLING RESOLUTIONS, AVERAGED OVER 107 MOLECULES.

Resolution	Min Diff.(%)	Max Diff.(%)	Avg. Diff.(%)
50×50	0.31	13.68	5.37
100×100	1.75	10.35	4.79
200×200	1.78	11.74	4.92
400×400	2.53	15.28	5.04

Table IV

MINIMUM, MAXIMUM AND AVERAGE DIFFERENCES BETWEEN VOLUMES MEASURED BY OUR METHOD AND MSMS [8] AT DIFFERENT UNIFORM SAMPLING RESOLUTIONS, AVERAGED OVER 107 MOLECULES.

Resolution	Min Diff.(%)	Max Diff.(%)	Avg. Diff.(%)
50×50	0.003	0.442	0.135
100×100	0.002	0.366	0.128
200×200	0.002	0.376	0.127
400×400	0.008	0.364	0.118

Table V

TIME IN SECONDS FOR VOLUME ESTIMATION ALONE, AS WELL AS FOR VOLUME AND SURFACE AREA ESTIMATION AND CAVITY DETECTION COMBINED, PERFORMED AT DIFFERENT SAMPLING RESOLUTIONS.

Resolution	Volume	Volume, surface area, cavity detection
50×50	0.02	0.04
100×100	0.08	0.19
200×200	0.27	1.40

general. It works with any type of closed surface for which a ray intersection procedure can be defined. Furthermore, the method directly benefits from any parallelism exploited by the ray casting framework. The same holds for scalability with model size.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a new unified framework for calculating the volume and other geometric properties of molecular surfaces. The core of our approach is a general-purpose ray casting framework, which enables us to compute highly accurate estimates of all kinds of geometric properties for all molecular models that can be efficiently intersected with rays. In particular, this holds for any triangulated molecular surface, but also for analytical surface models. Comparison of our technique with a well-established reference algorithm shows great accuracy and performance.

The presented method is general enough that a user will be able to apply it to almost any kind of model required (even in conjunction with more sophisticated concepts like clipping regions) without the need to modify the program. In addition, it is fast enough that it will not be the bottleneck in any of its application scenarios. The method will be fully integrated into our real-time ray tracing enabled molecular viewer and modeller BALLView [39], [40]. With this integration, all measurements can be efficiently performed on any displayed

representations through the click of a button, without the need to run any external programs.

In summary, we conclude that ray casting methods are very suitable to address problems in structural bioinformatics, apart from the visualization context they were originally designed for. Our future work will focus on numerical stabilization of the above described methods, exploring additional geometric features. Further work will be aimed at other kinds of surface representations, as our method is applicable to any representation that can be intersected with a ray. This, for instance, holds for many implicit molecular surface definitions (implicit SES, skin surfaces, etc.), where triangulation can be altogether avoided.

REFERENCES

- [1] L. Marsalek, I. Georgiev, P. Slusallek, A. K. Dehof, S. Nickels, H.-P. Lenhof, and A. Hildebrandt, "Real-Time Ray Tracing of Complex Molecular Scenes with BALLView," *Manuscript in preparation*, 2010.
- [2] M. Chavent, B. Levy, and B. Maigret, "Metamol: high-quality visualization of molecular skin surface." *J Mol Graph Model*, vol. 27, no. 2, pp. 209–216, Sep 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.jmgm.2008.04.007>
- [3] D. Whitley, "Van der Waals surface graphs and molecular shape," *J Math Chem*, vol. 23, no. 3–4, pp. 377–397, 1998.
- [4] B. Lee and F. M. Richards, "The interpretation of protein structures: estimation of static accessibility." *J Mol Biol*, vol. 55, no. 3, pp. 379–400, Feb 1971.
- [5] B. L. Greer, Jonathan; Bush, "Macromolecular shape and surface maps by solvent exclusion," *Biophysics*, vol. 75, pp. 303–307, 1978.
- [6] F. M. Richards, "Areas, volumes, packing and protein structure." *Annu Rev Biophys Bioeng*, vol. 6, pp. 151–176, 1977. [Online]. Available: <http://dx.doi.org/10.1146/annurev.bb.06.060177.001055>
- [7] M. L. Connolly, "Solvent-accessible surfaces of proteins and nucleic acids." *Science*, vol. 221, no. 4612, pp. 709–713, Aug 1983.
- [8] M. F. Sanner, A. J. Olson, and J. C. Spehner, "Reduced surface: an efficient way to compute molecular surfaces." *Biopolymers*, vol. 38, no. 3, pp. 305–320, Mar 1996.
- [9] B. B. Masek, A. Merchant, and J. B. Matthew, "Molecular skins: a new concept for quantitative shape matching of a protein with its small molecule mimics." *Proteins*, vol. 17, no. 2, pp. 193–202, Oct 1993. [Online]. Available: <http://dx.doi.org/10.1002/prot.340170208>
- [10] H. Edelsbrunner, "Deformable smooth surface design," *Discrete and Computational Geometry*, vol. 21, pp. 87–115, 1999.
- [11] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam, "Analytical shape computation of macromolecules: I. molecular area and volume through alpha shape." *Proteins*, vol. 33, no. 1, pp. 1–17, Oct 1998.
- [12] J. Ryu, R. Park, and D.-S. Kim, "Molecular surfaces on proteins via beta shapes," *Computer-Aided Design*, vol. 39, pp. 1042–1057, 2007.
- [13] P. W. Bates, G. W. Wei, and S. Zhao, "Minimal molecular surfaces and their applications." *J Comput Chem*, vol. 29, no. 3, pp. 380–391, Feb 2008. [Online]. Available: <http://dx.doi.org/10.1002/jcc.20796>
- [14] H. Gibson, K.D.; Scheraga, "Exact calculation of the volume and surface area of fused hard-sphere molecules with unequal atomic radii," *Molecular Physics*, vol. 62, p. 1247, 1987.
- [15] T. J. Richmond, "Solvent accessible surface area and excluded volume in proteins. analytical equations for overlapping spheres and implications for the hydrophobic effect." *J Mol Biol*, vol. 178, no. 1, pp. 63–89, Sep 1984.
- [16] O. V. Tsodikov, M. T. Record, and Y. V. Sergeev, "Novel computer program for fast exact calculation of accessible and molecular surface areas and average surface curvature." *J Comput Chem*, vol. 23, no. 6, pp. 600–609, Apr 2002. [Online]. Available: <http://dx.doi.org/10.1002/jcc.10061>
- [17] W. Fraczkiwicz, R.; Braun, "Exact and efficient analytical calculation of the accessible surface areas and their gradients for macromolecules," *Journal of Computational Chemistry*, vol. 19, pp. 319–333, 1998.
- [18] M. Totrov and R. Abagyan, "The contour-buildup algorithm to calculate the analytical molecular surface." *J Struct Biol*, vol. 116, no. 1, pp. 138–143, 1996. [Online]. Available: <http://dx.doi.org/10.1006/jsbi.1996.0022>
- [19] S. Bhat and E. O. Purisima, "Molecular surface generation using a variable-radius solvent probe." *Proteins*, vol. 62, no. 1, pp. 244–261, Jan 2006. [Online]. Available: <http://dx.doi.org/10.1002/prot.20682>
- [20] W. Heiden, M. Schlenkrich, and J. Brickmann, "Triangulation algorithms for the representation of molecular surface properties." *J Comput Aided Mol Des*, vol. 4, no. 3, pp. 255–269, Sep 1990.
- [21] H.-L. Cheng and X. Shi, "Guaranteed quality triangulation of molecular skin surfaces," in *IEEE Visualization*, 2004.
- [22] J. Ryu, R. Park, J. Seo, C. Kim, H. C. Lee, and D.-S. Kim, "Triangulation of molec surfaces," *Computer-Aided Design*, vol. 41, pp. 463–478, 2009.
- [23] W. Rocchia, S. Sridharan, A. Nicholls, E. Alexov, A. Chiabrera, and B. Honig, "Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: applications to the molecular systems and geometric objects." *J Comput Chem*, vol. 23, no. 1, pp. 128–137, Jan 2002. [Online]. Available: <http://dx.doi.org/10.1002/jcc.1161>
- [24] M. Petitjean, "On the analytical calculation of van der waals surfaces and volumes: Some numerical aspects," *Journal of Computational Chemistry*, vol. 15, pp. 507–523, 1994.
- [25] M. L. Connolly, "Computation of molecular volume," *Journal of the American Chemical society*, vol. 107, pp. 1118–1124, 1985.

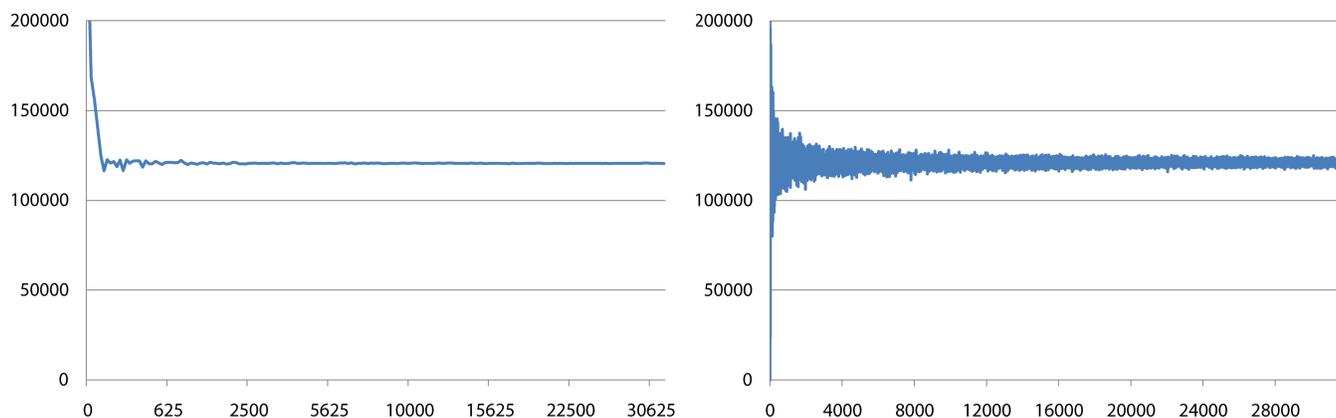


Figure 4. Graphs showing the convergence of our volume estimate for molecule 1b12 using uniform samples (left) and random samples (right). The horizontal and vertical axes are number of samples and the volume estimate, respectively. Uniform sampling converges at a faster rate.

- [26] F. M. Richards, "The interpretation of protein structures: total volume, group volume distributions and packing density." *J Mol Biol*, vol. 82, no. 1, pp. 1–14, Jan 1974.
- [27] E. Silla, F. Villar, O. Nilsson, J. L. Pascual-Ahuir, and O. Tapia, "Molecular volumes and surfaces of biomacromolecules via gepol: a fast and efficient algorithm." *J Mol Graph*, vol. 8, no. 3, pp. 168–72, 151, Sep 1990.
- [28] L. Willard, A. Ranjan, H. Zhang, H. Monzavi, R. F. Boyko, B. D. Sykes, and D. S. Wishart, "Vadar: a web server for quantitative evaluation of protein structure quality." *Nucleic Acids Research*, vol. 31, no. 13, pp. 3316–3319, Jul 2003.
- [29] A. S. Glassner, *An Introduction to Ray Tracing*. Morgan Kaufmann, 1989.
- [30] I. Wald, "Realtime Ray Tracing and Interactive Global Illumination," Ph.D. dissertation, 2004.
- [31] C. Benthin, "Realtime Ray Tracing on Current CPU Architectures," Ph.D. dissertation, Saarland University, 2006.
- [32] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*. New York, NY, USA: Academic Press, 1975.
- [33] J. Hammersley and D. Handscomb, *Monte Carlo methods*. New York, NY, USA: Halsted Press, 1964.
- [34] M. H. Kalos and P. A. Whitlock, *Monte Carlo methods. Vol. 1: basics*. New York, NY, USA: Wiley-Interscience, 1986.
- [35] S. Sonavane and P. Chakrabarti, "Cavities and atomic packing in protein structures and interfaces," *PLoS Comput Biol*, vol. 4, no. 9, p. e1000188, 09 2008.
- [36] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 2000.
- [37] O. Kohlbacher and H.-P. Lenhof, "Ball - rapid software prototyping in computational molecular biology," *Bioinformatics*, vol. 16, no. 9, p. 815824, 2000. [Online]. Available: <http://www.ball-project.org>
- [38] I. Georgiev and P. Slusallek, "RTfact: Generic Concepts for Flexible and High Performance Ray Tracing," in *IEEE/Eurographics Symposium on Interactive Ray Tracing 2008*, August 2008.
- [39] A. Moll, A. Hildebrandt, H.-P. Lenhof, and O. Kohlbacher, "Ballview: a tool for research and education in molecular modeling." *Bioinformatics*, vol. 22, no. 3, pp. 365–366, 2006. [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/bti818>
- [40] "The BALLView Website," <http://www.ball-project.org>.