



Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

B. KOZLÍKOVÁ, M. KRONE, M. FALK, N. LINDOW,
M. BAADEN, D. BAUM, I. VIOLA, J. PARULEK,
AND H.-C. HEGE

Visualization of Biomolecular Structures: State of the Art

The manuscript will appear in a slightly revised version in the EuroVis STAR 2015 proceedings.

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Visualization of Biomolecular Structures: State of the Art

B. Kozlíková¹, M. Krone², M. Falk³, N. Lindow⁴, M. Baaden⁵,
D. Baum⁴, I. Viola^{6,7}, J. Parulek⁶, and H.-C. Hege⁴

¹Faculty of Informatics, Masaryk University, Czech Republic

²Visualization Research Center, University of Stuttgart, Germany

³Immersive Visualization Group, Linköping University, Sweden

⁴Department of Visual Data Analysis, Zuse Institute Berlin, Germany

⁵Laboratoire de Biochimie Théorique, UPR 9080 CNRS, France

⁶Department of Informatics, University of Bergen, Norway

⁷Institute of Computer Graphics and Algorithms, TU Wien, Austria

Abstract

Structural properties of molecules are of primary concern in many fields. This report provides a comprehensive overview on techniques that have been developed in the fields of molecular graphics and visualization with a focus on applications in structural biology. The field heavily relies on computerized geometric and visual representations of three-dimensional, complex, large, and time-varying molecular structures. The report presents a taxonomy that demonstrates which areas of molecular visualization have already been extensively investigated and where the field is currently heading. It discusses visualizations for molecular structures, strategies for efficient display regarding image quality and frame rate, covers different aspects of level of detail, and reviews visualizations illustrating the dynamic aspects of molecular simulation data. The survey concludes with an outlook on promising and important research topics to foster further success in the development of tools that help to reveal molecular secrets.

1 Introduction

Interactive molecular visualization is one of the oldest branches of data visualization [105, 48], with deep roots in the pre-computer era. This paper reviews interactive visualization of *biomolecular* structures – the subfield that developed most during the past two decades.

First, let us characterize the objects of interest. Ordinary matter consist of atoms and molecules, which in turn embody protons, neutrons and electrons. The protons and neutrons are bound together by nuclear forces, forming the nuclei of the atoms. The positively charged nuclei attract negatively charged electrons; due to quantum mechanical effects the particles do not collide, but the electrons surround the nuclei in defined distances, comprising stable and electrically neutral *atoms*. These are the smallest units of a chemical element. The electrons in an atom are organized in orbitals, i.e.,

regions of space, in which electrons stay with high probability. Each atomic orbital can contain up to two electrons, whose energy, angular momentum and magnetic moment determine the orbital's shape and location. In a simplified model, atomic orbitals are arranged in layered shells and sub-shells. The outer electrons of two atoms can interact and form molecular orbitals, potentially creating a *chemical bond* between the atoms.

Bonds are classified as being either strong (covalent, ionic, and metallic bonds) or weak (dipole-dipole interactions and hydrogen bonds). Due to strong bonds sets of atoms are held together, forming tight entities like molecules, ionic salts, and metals. A *molecule* thus is a structure composed of nuclei, defining the atom positions, and their core electrons (inner electron shells); these nuclei are held together by an outer electronic shell (valence shell), composed of molecular orbitals. Molecules are the smallest units of a compound, i.e., of a pure chemical substance. Molecules playing an active role in living systems are called *biomolecules*. These include large molecules (macromolecules) such as proteins, polysaccharides, lipids, DNA and RNA, as well as small molecules such as metabolites. Weak bonds occur inside molecules as well as between molecules. They are critical in maintaining the 3D structures of biomolecules, in forming larger entities (molecular complexes) and in binding molecules specifically but transiently, creating thereby the basis of many biological processes.

The primary purpose of *molecular visualization* is to support our understanding of the rich, complex material world, by making molecular structures, their properties and their interactions apprehensible and intelligible. In addition it aims at supporting the 'rational' design of new molecules, such as pharmaceutically active compounds, or customized substances with specific properties. The subfield *biomolecular visualization* deals with the graphical depiction of the structure, interaction and function of biomolecules, biomolecular complexes, molecular machines and entire biological functional units that occur in biological cells. Additionally, it complements the toolset of bioinformatics by providing means for integrated visual analysis of sequence and structure data, e.g. for elucidating phylogenetic trees; examples are viewers displaying families of closely related structures, mutants, and conserved structural motifs.

Forerunners of today's visual representations of atoms and molecules are *hand-drawn depictions* and *physical models*. Pictorial representations have been used, e.g., by Kepler (1611) [82] and Huygens (1690) [73], centuries before 1808, when Dalton published the modern, but still pre-quantum formulation of atomic theory [36]. In these groundbreaking works, atomic arrangements were illustrated, displaying atoms as spheres. Van der Waals, when deriving the equation of state for gases and liquids (1873) [175], saw the necessity of taking into account the molecular volume as well as attracting intermolecular forces; he computed from experimental data the volume occupied by an individual atom or molecule. From now on, approximate atomic radii for several chemical elements were known and used in depictions. Also physical models of molecules, both static and dynamic, have been used for visualization purposes. They allowed to correlate mechanical, optical, electrical, magnetic, and thermal properties with the atomic structure and to reproduce atomic and molecular movements. For a survey covering the period 1880–1960 see [155].

With the emergence of increasingly elaborate atomic models by Thompson, Rutherford, Bohr, and Sommerfeld in the early 19th century, more detailed visualizations became necessary, culminating in detailed depictions of complex atoms in the popular book of Kramers and Holst [89], showing the elliptic orbits of the electrons in the Bohr-Sommerfeld model. However, in these years it became clear that atoms and molecules are of truly quantum nature. Quantum physics, however, seems to be intrinsically *non-visualizable*. One of several reasons is that no (mental) image exists that

simultaneously represents the corpuscular and wave-like character of particles. According to Heisenberg’s uncertainty relation, an electron cannot be considered to have an exact location in its orbital, i.e. its trajectory is not defined [71]. Instead, according to Born [17], an electron’s position is described by a probability distribution, given by the absolute square of Schrödinger’s complex wave function Ψ . The evolution of Ψ for a system of N quantum particles, described by the time-dependent Schrödinger equation [148], happens not in real 3-dimensional space, but in $3N$ -dimensional space of all particles’ coordinates—which poses a further challenge to visualization. Regarding visualization of fully quantum physical systems only very limited work is available; examples are [167, 18].

Fortunately, research revealed that molecular systems can be described *classically* to a good approximation, if no covalent bonds are newly formed or broken, and if the system’s behavior does not depend sensitively on fine-tuned energy values. In *molecular dynamics (MD) simulations*, no molecular orbitals are computed; atoms are treated as classical objects that move under the influence of artificial multi-body forces (‘force fields’) that mimic quantum effects. Due to the strong repulsion between neutral atoms and molecules, atoms can be considered approximately as ‘hard’ spheres; this means, atoms are fully characterized by their mass, radius and the multi-body forces they exert on other atoms, ‘inner’ electronic degrees of freedom are neglected. The majority of MD simulations, particularly of biomolecular structures, is performed using this ‘classical’ approximation. The depiction of van der Waals spheres thus was one of the starting points of modern molecular computer graphics, beginning with the work of Lee and Richards (1971) [102]. This work has been continued, now for more than four decades, with the invention of further types of molecular surfaces representing the spatial accessibility of molecules.

However, some applications require to model at least a few degrees of freedom quantum mechanically, using methods of quantum chemistry (for an elementary introduction see, e.g., [70]). Then the data to be visualized are expectation values of physical variables, like, e.g., electron and nuclear densities or fluxes, describing, e.g., equilibrium geometries and reaction energetics. On the visualization side the depiction of (multi-)fields is required, see, e.g. [12, 6]. Topological visual analysis of such fields provides information about the spatial domains attributed to individual atoms [9] and the classification of bonds [152, 67]. Also visualization of the resulting fuzzy molecular surfaces using volume rendering has been proposed [83].

In the next section, the basics of biomolecular data are outlined, including data sources. Section 3 introduces a taxonomy of the literature about molecular visualization covered by this report and gives an overview of the structure of the rest of the paper. The report is concluded by a brief overview of molecular visualization tools (Section 7) and anticipated future challenges for the field (Section 8).

2 Molecular Data

This section introduces the input data, mostly formed by biomolecules, along with their composition and basic properties. Moreover, we discuss the most common sources of molecular structures, which partially enable the study of their dynamic behavior as well.

2.1 Biomolecules

Biomolecules usually carry out important functionality including enzymatic catalysis, coordinated motion, mechanical support, immune protection, generation and transmission of nerve impulses, and reproduction [162]. Some of these molecules are rather large entities and are, therefore, referred to as macromolecules. Others are building blocks of complex structures such as membranes. The majority of small biomolecules takes an active role in the metabolism of an organism and are hence called metabolites. Below, the most important species of biomolecules are briefly introduced.

The building blocks of *nucleic acids* are nucleotides, which are composed of a nucleobase, a sugar, and a phosphate group. The main difference between deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) is that the sugar in DNA is a deoxyribose instead of a ribose as in RNA. Further, one of the four bases occurring in DNA, thymine, is replaced by uracil in RNA. DNA usually forms the characteristic double helix of two single DNA strands first identified by Watson and Crick [177]. In contrast, RNA is single-stranded and typically forms very complex structures by folding onto itself. DNA stores the genetic code including the information about the composition of the proteins; both DNA and associated proteins may be epigenetically modified to regulate gene expression. During protein synthesis, the respective part of the DNA is transcribed into RNA, which is in turn translated into the amino acids that form the protein.

Proteins are macromolecules consisting of one or more chains of amino acids. Different proteins have diverse functions like replication of DNA, catalyzing chemical reactions, or transport of other molecules. The amino acids forming the protein chain are connected via peptide bonds. This chain is called the protein's primary structure. The amino acid chain folds into an energetically favorable configuration stabilized by intramolecular interactions, such as hydrogen bonds. The folding introduces patterns to the protein chain called secondary structure. The two most common secondary structure elements are the α -helix [125] and the β -sheet [124], which are connected by loops and unstructured parts called random coil. The correct folding of the chain is important for the proper function of most proteins. The three-dimensional arrangement of the secondary structure of the protein chain is called tertiary structure. Two or more folded chains can form a functional complex called quaternary structure. Note that in the visualization literature, the term secondary structure may be used synonymously for tertiary and quaternary structure (e.g., [176]). In addition, proteins may undergo a post-translational modification during their maturation.

Lipids and *lipid membranes* are ubiquitous in biological systems as they delineate the compartments of the cell, control entry and transport, and harbor important membrane proteins. In addition to lipids, proteins and nucleic acids, cells contain *sugar molecules* carrying out crucial biological functions and storing energy. Sugars may attach to proteins or lipids and form extremely complex polymers, the polysaccharides. Many *small molecules*, *metabolites*, and *ions* are further central ingredients of life [57]; they are actually frequently present and important in structural data. A few examples include energy-providing ATP, electron-transporting NAD, and other prosthetic groups.

2.2 Molecular Structure Acquisition

In-vitro experiments provide a key resource for molecular structural data based on three techniques. *X-ray crystallography* [179] potentially leads to the highest resolution data when crystals can be obtained. *Nuclear magnetic resonance spectroscopy*

(NMR) [180] determines structural ensembles rather than a single structure. *Cryo-electron microscopy* (cryo-EM) [172] requires an image-based reconstruction currently limiting the resolution, yet it allows the determination of large structures.

Molecular simulation is a useful method to study the dynamic behavior of previously determined molecular structures. It allows scientists to study the effect of different parameters like varying temperature, different solvents, and the interaction with other molecules. The most frequently used methods are Monte Carlo (MC) sampling and molecular dynamics (MD) simulations. An overview of these methods can be found in the textbooks by Frenkel and Smit [153] and Schlick [146]. Both methods usually ignore quantum mechanical effects or incorporate such effects only through the molecular force field being used. Hybrid MC methods have been developed to combine the merits of both methods. If the molecular systems to be simulated become very large (several million to billion of atoms), these simulation methods are computationally very expensive to simulate the system for a relevant time interval of milliseconds to seconds. Although Shaw et al. [149] are now capable of running ribosome-sized simulations of a few million atoms at multiple microseconds per day, in most cases it is still necessary to abstract from atomic resolution and move to coarse-grained models. In these models, groups of atoms instead of single atoms are considered as the smallest unit. Depending on the molecular systems of interest, several coarse-grained models can be adopted (see, e.g., [29]). If a process is mainly controlled by diffusion, Brownian Dynamics (BD) is often used as a complementary approach to MD [3].

The results of molecular modeling and simulation methods are trajectories of coordinates of particles. In the case of all-atom simulations, these particles are atoms while for coarse-grained simulations, each particle represents the center of mass of a molecule or a group of atoms.

In contrast to the molecular simulation techniques mentioned above, normal mode analysis (NMA) calculates large-amplitude molecular motions without simulating the motion of a molecule [10]. It is much faster than classical molecular simulation and, thus, allows the study of large-scale macromolecular motions taking place at a long time scale.

Recently, Johnson et al. developed a semi-automatic modeling tool called cellPack [76] that computes a packing of molecules to form comprehensive models of very complex molecular systems up to mesoscopic length scales.

Another source of data are biochemical reaction models, which can be categorized roughly into two classes: kinetic models and particle-based models. Kinetic models are typically described by pathway networks augmented with spatial information at times. In contrast, the focus of particle-based models lies on the action and interaction of individual agents, i.e., the particles. An agent is assigned with a set of rules of how to behave in a certain environment and how to interact with other agents, i.e., other molecules. Popular frameworks for simulating cellular environments with particles include *MCell* [157], *ChemCell* [129], and *Smoldyn* [7], covering membrane interaction, diffusion, and particle-particle reactions. The computational cost of agent-based simulations is usually very high and time-consuming compared to kinetic models. Another efficient method to study biochemical reaction models is stochastic simulation [53]. Recently it was shown [137] that the chemical master equation and the reaction-diffusion master equation, both underlying stochastic simulations, can be efficiently sampled on the GPU speeding up the computation up to two orders of magnitude.

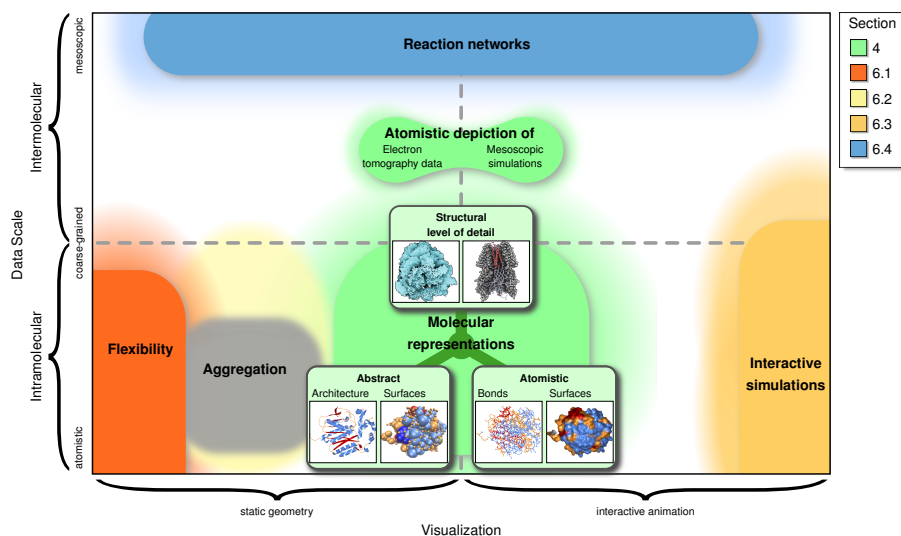


Figure 1: Illustrated taxonomy of the literature about molecular visualization covered by this report. Miniatures created by CAVER Analyst [88] and taken from [121, 171].

3 Taxonomy

Figure 1 depicts the taxonomy that is used to classify the methods covered by this report. We distinguish between four major areas shown as quadrants in the figure. These quadrants are defined by the type of visualization along the horizontal axis and the data scale on the vertical axis.

The type of visualization can be subdivided into showing a *static geometry* (left side) or depicting an *interactive animation* (right side). Visualizing static geometry results in a still image. Such a still image can nonetheless show dynamic properties directly related to the temporal domain or derived attributes. The interactive animation on the other hand focuses on the interactive playback to further emphasize the dynamics of the data. Instead of showing a pre-rendered movie, the animation is computed and shown on demand. In both cases, the visualization typically allows for interactive adjustment of parameters like camera settings by the user.

The vertical axis corresponds to the scale of the underlying data that is visualized. Although being continuous, this axis can be divided into two major areas with respect to molecular visualization. The *intramolecular scale* ranges from atomistic data on the atomic scale to coarse-grained molecular models. The *intermolecular scale* covers coarse models up to the mesoscopic level where entire molecules are considered as a single entity. The actual scale of the data mostly depends on the data acquisition, e.g., molecular structures obtained by NMR or results of mesoscopic intracellular simulations. Please note that coarse data might be enriched in the visualization to add more details. One example of such an augmentation is the replacement of structural data on the intermolecular scale with details on the atomistic scale, i.e., individual atoms [106, 47]. Furthermore, additional bioinformatics data like phylogenetic trees and other biomolecular information can be included as well.

The colored areas in Figure 1 correspond to the various concepts discussed in the subsequent sections. Their positions coincide with the type of visualization and data

scale where the respective methods and algorithms are typically applicable to. Molecular representation models (green) are described in Section 4. These representations can be divided into atomistic models (Section 4.1), illustrative and abstract models (Section 4.2), and structural level of detail (Section 4.3). They can be applied to visualize static and dynamic attributes on the intramolecular scale. One exception is the depiction of atomistic detail on the intermolecular scale, which utilizes the enrichment described above (cf. Section 4.3). The remaining areas can be summarized under the term of visualization of molecular dynamics (Section 6). This includes the visualization of flexibility (red, Section 6.1), volumetric representations and aggregation (yellow, Section 6.2), interactive and steered simulations (orange, Section 6.3), and the visualization of molecular reactions (blue, Section 6.4). The techniques for molecular rendering described in Section 5 are not included in the taxonomy since they are generally applicable to the majority of molecular visualizations. Finally, Section 7 gives an overview on existing tools and systems available for molecular visualization with respect to this taxonomy.

4 Molecular Representation Models

In chemistry, many three-dimensional molecular models have been developed that show different attributes of the depicted molecule. The choice of the molecular model used for data visualization depends on the intended analysis task. The models can be classified into atomistic ones (Section 4.1) and abstract ones (Section 4.2), as is shown in the taxonomy illustrated in Figure 1. Large molecular systems are often depicted using level of detail visualizations (Section 4.3), which include the continuous representations as defined by Goodsell [58] that simplify the atomic details.

4.1 Atomistic Models

Atomistic models directly depict atoms of a molecule. The atomic structure always plays an essential role in determining molecular properties. The atomistic representations model discrete entities interacting through pair-wise forces and are usually used in molecular systems consisting of up to millions of atoms. The atomistic models can be further classified into models that focus on the bonds between atoms and surface models, which illustrate the interface between a molecule and its environment.

In traditional interactive molecular graphics, molecular models were typically triangulated since GPUs are designed for fast triangle rendering. To achieve a reasonable quality, however, often many triangles are required, which can impede interactivity. Since many models can be decomposed into simple implicit surfaces, e.g., spheres and cylinders, modern GPU-based glyph ray casting presented by Gumhold to render ellipsoids [66] became more efficient. The general idea is to render a projection of a primitive that bounds each implicit surface (i.e., glyph). Then, for each fragment of said primitive, the intersection of the view ray with the implicit surface is computed in the fragment shader. Reina and Ertl used a combined ray casting of spheres and cylinders to visualize mono- and dipoles in MD data [132]. Sigg et al. [151] formulated a general concept for ray casting arbitrary quadrics on the GPU. GPU-based ray casting can still be seen as the current state-of-the-art. It enables rendering a massive number of simple surface patches in real-time with pixel-perfect quality and for any zoom level.

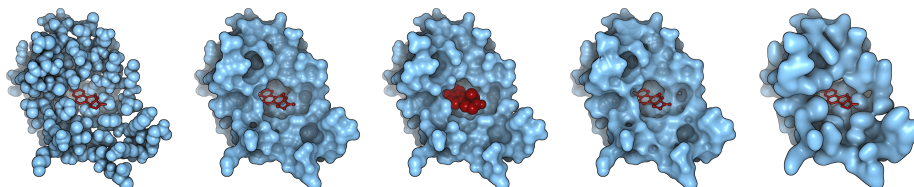


Figure 2: Comparison between different molecular surfaces of the protein isomerase (PDB ID: 1OGZ). From left to right: vdW surface, SES with probe radius 1.4 Å, LES for equilenine, MSS with shrink factor 0.35, and Gaussian surface with standard deviation equal to the atom radius. The ligand equilenine (red) is shown as stick, ball-and-stick, or vdW surface, respectively. In all examples, depth cueing in combination with screen-space ambient occlusion is applied and silhouettes are shown.

4.1.1 Bond-centric models

Visualizing chemical bonds between atoms helps to understand and predict many chemical properties of the given molecule. Bond-centric models that display the chemical bonds between individual atoms of the molecular system were designed for this purpose. The most often used bond-centric model visualizing only bonds is called *licorice* or *stick* model. The bonds can be augmented with the atoms forming these bonds, which results in a representation called *ball-and-stick*, which is one of the oldest and most often used structural representations.

The simplest representation of bonds is the *lines* model. More sophisticated visualizations represent the bonds by cylinders and atoms by spheres. As described above, modern GPU-based ray casting techniques are much more efficient and achieve higher visual quality than traditional triangle-based rendering for these implicit objects. However, most of the modern techniques for bond representation and van der Waals representation are descendants of techniques and software tools that came out in the late 80s and early 90s [49, 115].

Chavent et al. [25] introduced a novel representation called *HyperBalls*. Instead of the traditional stick representation of bonds, it smoothly connects atom spheres by hyperboloids. Hyperboloids can be defined by a cubic equation, which makes them suitable for GPU-based ray casting.

4.1.2 Surface Models

Space-filling Model and Van der Waals Surface. The simplest and probably most often used molecular model is the *space-filling* or *calotte* model. Here, each atom is represented by a sphere whose radius is proportional to the atomic radius, e.g., covalent radius, of the respective element. The surface is then defined as the outer surface of the union of all atom spheres (blue spheres in Figure 3). The *van der Waals (vdW) surface* [135, 52] is a space-filling model where the radius of the atom spheres is proportional to the van der Waals radius (Figure 2). This surface shows the molecular volume, that is, it illustrates how much space a molecule occupies. The vdW surface is the basis for most other molecular surface representations. Nowadays, GPU-based ray casting of the vdW spheres is the fastest way to interactively visualize the surface for several million of atoms [64]. Recently, further techniques were proposed to handle even larger data sets (see Section 4.3).

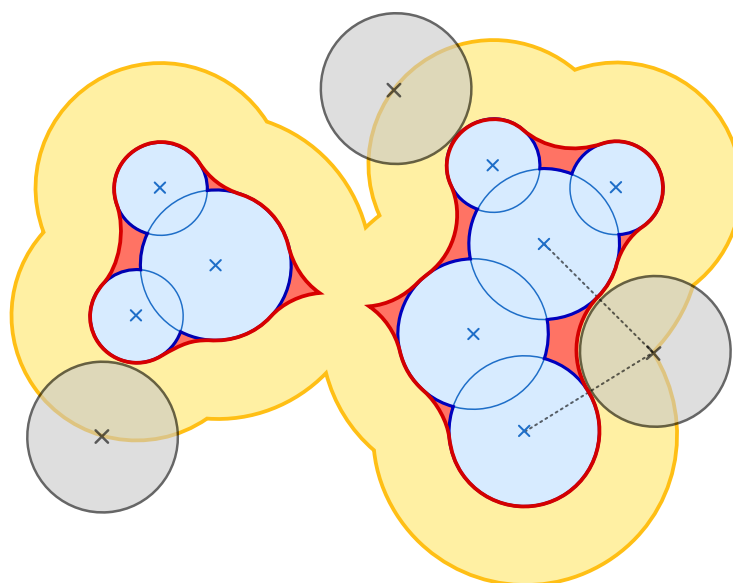


Figure 3: 2D schematic of vdW surface (blue), SAS (yellow), and SES (red). The SAS and SES are defined by a spherical probe (gray) that rolls over the vdW surface.

Solvent Accessible Surface. Lee and Richards defined one of the first extensions to the vdW surface, which later became known as the *solvent accessible surface* (SAS) [102]. The idea of this surface is to show all regions of a molecule that can be accessed by a solvent molecule. To simplify the computation, the solvent molecule is approximated by a single sphere—the probe. The SAS is described by the center of the probe while rolling over the vdW surface, as shown in Figure 3. During this process, the probe always touches the vdW surface but never penetrates it. All points outside the surface can be geometrically accessed by the center of the probe and, thus, probably also by the solvent. Consequently, all atom spheres contributing to the SAS are accessible to a molecule with radius equal to or smaller than the probe radius. This makes the SAS feasible for analyzing possible binding partners or transport channels. The disadvantage of the SAS, however, is that it does not faithfully show the molecular volume since the molecule is inflated. This can lead to intersections with other molecules, e.g., when visualizing a molecular simulation. The SAS is identical with the vdW surface where each vdW radius is extended by the radius of the probe. All visualization techniques for the vdW surface can also be used to render the SAS.

Solvent Excluded Surface. In 1977, Richards [135] defined the first smooth molecular surface (see Figure 2) based on the idea of the SAS. Instead of taking the center of the probe that rolls over the atoms, he suggested to use the outer shell of the probe (see Figure 3). This combines the advantages of both previous surfaces, the better size representation of the vdW surface and the accessibility visualization of the SAS. Greer and Bush gave an alternative definition [60], which is equal to the one of Richards. They defined the surface as the topological boundary of the union of all possible probe spheres that do not penetrate any atom of the molecule. Their work coined the commonly used term *Solvent Excluded Surface* (SES). The SES is composed of three types of patches. Convex spherical patches occur where the probe touches exactly one atom;

toroidal (or saddle) patches are tracks where the probe touches exactly two atoms; concave spherical patches occur where the probe lies in a fixed position, touching exactly three atoms. At the patch boundaries, where two or more patches fit together, the surface is C^1 -continuous, which means that the SES is smooth. However, the surface can contain self-intersections, which are also called singularities [143]. At these intersections the surface has sharp edges and is only C^0 -continuous. Two different types of self-intersections can occur when the atoms lie too far away from each other. The first type is the self-intersection of toroidal patches. This type occurs when the probe rolls between two atoms and intersects the axis of revolution through the two atom positions, thereby creating a spindle torus. The second type occurs when two or more concave spherical patches intersect each other.

The algorithms for computing the SES can be divided into two categories. The first comprises all methods that compute the surface by discretizing the space of \mathbb{R}^3 . These approaches usually compute a discrete scalar field from which an isosurface is extracted either by triangulation via Marching Cubes [109] or by direct isosurface ray marching. Two of the fastest approaches in this research area were presented by Can et al. [20] and Yu [181]. Although these algorithms are typically easy to implement, the computation time and memory requirements increase cubically with the grid resolution. The second category contains all methods that compute an analytical representation of the surface by determining the implicit surface equations of all patches. In 1983, Connolly [31] presented the equations to compute the SES analytically and the first algorithm. Varshney et al. [173] proposed a parallel algorithm based on the computation of an approximate Voronoi diagram. Edelsbrunner and Mücke [43] introduced *alpha-shapes* that can be used to compute the SES. Sanner et al. [143] presented the *reduced surface* (RS) algorithm. This algorithm is very efficient but iterative and, thus, not easily parallelizable. The RS can be updated partially in order to support dynamic data [142]. In 2009, Krone et al. [93] achieved interactive frame rates for dynamic molecules with a few thousands of atoms using an optimized implementation of the RS algorithm. In 1996, the same year Sanner et al. presented their reduced surface algorithm, Totrov and Abagyan [169] proposed the *contour-buildup* (CB) algorithm. It directly computes the track of the probe on each atom surface and is, therefore, an embarrassingly parallel problem. Lindow et al. [108] presented a parallel CB algorithm using OpenMP, which allowed the user to visualize dynamic molecules with up to 10^4 atoms on 6 core systems. Krone et al. [94] parallelized the CB algorithm for GPUs, which further accelerated the SES computation and enabled the interactive visualization of dynamic molecules with up to 10^5 atoms. These two methods are currently the fastest analytical techniques to compute the SES.

For visualization purposes, the SES was traditionally tessellated. Examples for two very accurate tessellations are the one by Sanner et al. [143] and the one by Laug and Borouchaki [98]. Later, Zhao et al. [182] proposed a triangulation that approximates the patches by spline surfaces to simplify the triangulation. One of the fastest methods was proposed by Ryu et al. using subdivision surfaces [140]. Their approach, however, is not able to handle all possible singularities.

Triangulating the SES is computationally expensive and takes usually seconds for mid-sized proteins. In 2009, Krone et al. [93] thus used GPU-based ray casting to render three types of surface patches. As mentioned above, it not only gives pixel-perfect image quality but is also much faster, even though quartic equations have to be solved. Krone et al. also handled the self-intersections of the SES patches using ray casting. Lindow et al. [108] presented a slightly improved ray casting that uses the geometry shader to optimize the rasterization of primitives. This accelerated the rendering by

approximately 30%. To optimize the ray casting performance, the parts of the convex spherical patches that lie inside the SES were not clipped in the previous methods. Hence, the surface could be only visualized opaque or with a simple blending of the front face. Semi-transparent or clipped visualizations, however, require a complete clipping of these patches. A solution for this was described by Kauker et al. [81]. Ray casting is currently the fastest techniques to visualize the SES while also offering the highest image quality.

In 2012, Parulek and Viola presented the first ray casting of the SES that does not need a pre-computation of the analytical description of the surface [122]. They use a modified sphere tracing and directly compute the implicit description of the surface based on the local neighborhood of the ray. This enables the direct visualization of the SES for dynamic molecular data. However, due to the complexity of this extended ray casting, interactive frame rates are only achieved for molecules up to 2,000 atoms. The technique also offers a level of detail strategy that improves the rendering performance, but can lead to pixel artifacts, e.g., at singularities and patch boundaries. Details can be found in the STAR by Patane and Spagnuolo [123] on geometric and implicit modeling for molecular surfaces.

Decherchi and Rocchia presented a combination of triangulation and ray casting [38]. The algorithm computes the analytical description of the SES and performs a ray casting along a 3D grid from which the surface is triangulated using Marching Cubes. Although they could accelerate the triangulation of the SES, the overall speed and visual quality cannot compete with direct ray casting.

Molecular Skin Surface. Edelsbrunner presented a new smooth surface for a finite set of input spheres, called *skin surface* [42]. Its shape depends on a single parameter $s \in (0, 1] \subset \mathbb{R}$, the shrink factor. The *molecular skin surface* (MSS) is the application of the skin surface to the vdW spheres of the atoms. The main advantage of the MSS over the SES is that the surface is completely C^1 -continuous (see Figure 2). Furthermore, it can be decomposed into patches of quadrics. However, the MSS has no biophysical background. Kruithof and Vegter [96] presented a topology certified tessellation approach for the MSS. Cheng and Shi [26] developed a triangulation algorithm that achieves a higher quality but has the disadvantage that it is very time consuming. A very fast triangulation was presented by Decherchi and Rocchia [38] following the same strategy as their SES approach. However, it does not necessarily preserve the full surface topology. To achieve a fast, high-quality visualization, Chavent et al. [22] presented the first GPU-based ray casting to render the MSS. The long run times of their implementation for the construction of the MSS, however, prevented the use for dynamic molecular data. In 2010, Lindow et al. [108] presented an accelerated computation using the same idea that Varshney et al. [173] applied to compute the SES. They also optimized the ray casting of the MSS. As result of both improvements, interactive MSS visualization of dynamic molecules with a few thousand atoms became possible.

Ligand Excluded Surface. The *ligand excluded surface* is a generalization of the SES (see Figure 2). It was recently proposed by Lindow et al. [107]. In contrast to the SES, the LES does not approximate the ligand by a sphere but uses the full geometry and dynamics of the ligand’s vdW surfaces. Thus, the surface shows a geometrically correct accessibility for a specific ligand. However, the analytical computation of the LES is difficult. For this reason, Lindow et al. proposed an algorithm to compute the surface by discretizing the possible ligand positions, orientations, and dynamics. While

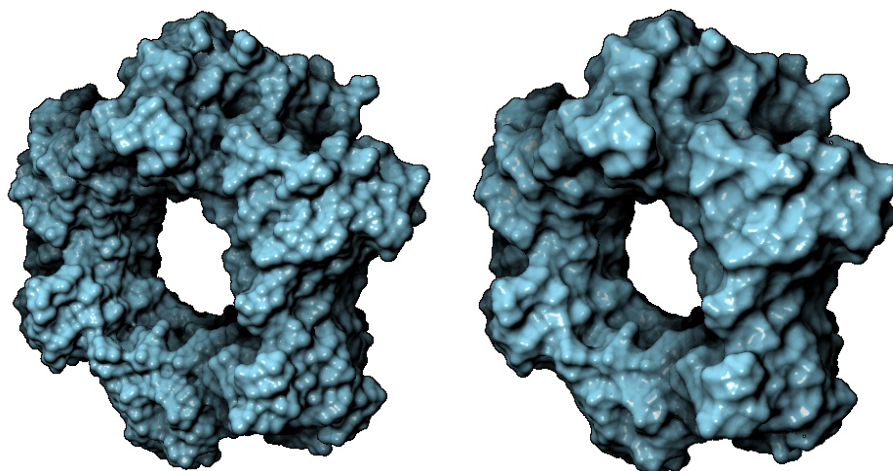


Figure 4: Molecular surface for proliferative cell nuclear antigen (PDB ID: 4D2G) represented by an implicit model defined by blending scheme [120]. The images show the result for different solvent radii: 1.4 Å (left) and 2.2 Å (right).

the LES provides the most accurate accessibility for a specific ligand, its computation takes several minutes for mid-sized proteins and a reasonable surface quality. Thus, if interactivity is required, the SES is favorable. The LES should be favored if a more detailed view of a static molecule is needed.

Convolution Surface Models. Blinn [16] introduced implicit modeling as an approximation of the molecular surface in 1982. He proposed the use of a Gaussian convolution kernel (see Figure 2) in order to blend atom potentials to describe the electron density function. The resulting surface is commonly known as *Metaballs*, *blobby surfaces*, or *convolution surfaces* [174]. Such a summation-based model, however, generally lacks information of the correlated solvent molecule. Therefore, Grant and Pickup [59] determined the parameters for the Gaussian-based model to mimic the volume and solvent accessible surface area (SASA) for different solvent probe sizes.

There are several other kernels mentioned in the literature that can be used as alternative kernel functions [150], i.e., avoiding computationally expensive exponential functions. One of the main advantages of kernel-based models is the simplicity of representation and model evaluation. For instance, the function to be evaluated has linear time complexity and the final formula can be expressed analytically. In 2013, Parulek and Brambilla [120] proposed another implicit model with linear complexity although its definition is not purely analytical compared to the Gaussian model for instance. On the other hand, it resembles the SES more closely than the kernel-based approaches (Figure 4). The main reason lies in the fact that the implicit function evaluation incorporates the solvent, represented by a sphere of a specific radius. An implicit space mapping is then exploited to approximate the circular distance to individual atoms.

In 2008, Kanamori et al. [80] proposed an efficient technique for ray casting the kernel-based models. It employs Bezier clipping to quickly compute an intersection between a ray and the surface. The GPU implementation exploits depth peeling to retrieve contributing spheres for the actual ray segment, where the iso-surface point is then evaluated through the Bezier clipping technique. To further speed-up the algo-

rithm, Szecsi and Illes [164] suggested to employ *fragment linked lists* or an *A-Buffer* to avoid the multi-pass rendering required by depth peeling.

In order to visualize models based on implicits, they are often discretized on a regular grid prior to rendering. Subsequently, a triangle mesh can be extracted for rendering, e.g., using Marching Cubes. However, when dealing with complex shapes such as molecular surfaces, a very fine-grained tessellation is needed for a fully detailed surface representation. To remove this limitation, Krone et al. [95] proposed an interactive visualization method to extract and render a triangulated molecular surface based on Gaussian kernels. They efficiently exploited GPGPU capabilities to discretize the density field, which is then processed by a GPU-accelerated Marching Cubes algorithm. The rendering performance depends on the resolution of the density grid as well as on the number of atoms. Their method achieves interactive frame rates even for molecules counting millions of atoms due to the high degree of parallelism and is currently among the fastest molecular surface extraction algorithms.

4.2 Illustrative and Abstract Models

Apart from molecular models that show a direct representation of the atoms of a molecule, several abstract models have been established. An abstract model might illustrate a special feature of the molecule, which is not or at least not clearly and easily discernible in an atomistic model. Abstract models can also lead to sparse representations, which might be easier to understand or reduce occlusion. Such a representation can for example be useful for a very large molecular complex, for which often not the individual atoms but the overall shape is important.

4.2.1 Representations of Molecular Architecture

Very early on, the conceptualization of complex macromolecular assemblies inspired scientists to simplify computer graphics images representing these entities. Visual abstraction of the molecular architecture often shows important structural features more clearly than a full-detail atomistic representation [117], e.g., using abstractions for molecular subunit structures [118]. Goddard and Ferrin alternatively refer to such abstractions as multiple levels of detail that match the underlying structural hierarchy of molecular assemblies [54]. As our understanding of biological structures progresses, the need for new abstractions may arise such as was the case for representing the bases of nucleic acid polymers and, more recently, carbohydrates.

In 1981, Richardson [136] introduced the *cartoon* representation for proteins, which depicts the secondary structure as ribbons and arrows. Since then, a variety of cartoon renderings have been developed that vary the graphical appearance, e.g., using straight cylinders for helices (see Figure 5). A current challenge is to improve the efficiency for the interactive visualization of large proteins. This can for example be achieved by mesh-refinement techniques at the software level [68] or by moving to GPUs at the hardware level. Several GPU implementations were proposed, starting with Krone et al. [92] comparing CPU, hybrid CPU/GPU, and full GPU implementations that exploit the geometry shader. Although with the available graphics hardware at that time the best performance was achieved with the CPU implementation, this might be no longer the case due to recent GPU developments. Using a hybrid CPU/GPU approach, Wahle and Birman report a near 3-fold speed-up for their cartoon implementation [176]. New variants of helix abstractions, with the aim to map simulation analysis data onto

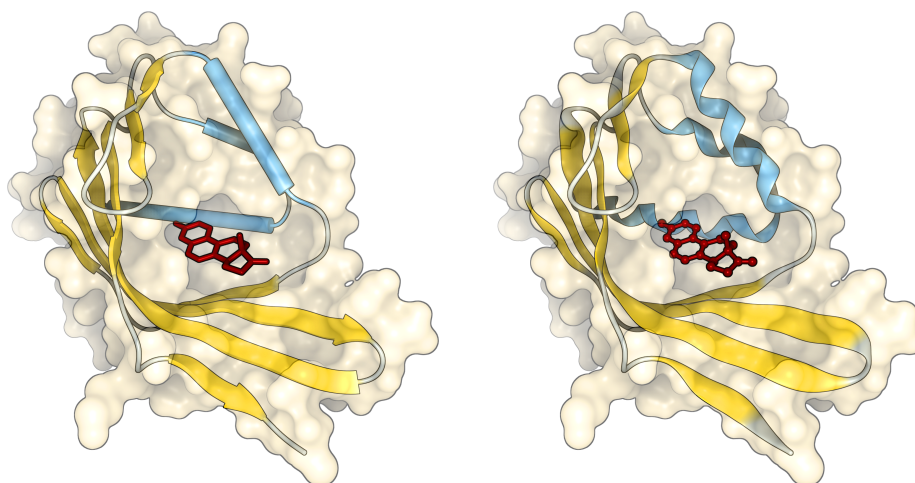


Figure 5: Two possible cartoon renderings of the same protein (PDB ID: 1OGZ). Left: Ribbon-shaped arrows show the direction of the amino acid chain for the β -sheets while the α -helices are stylized as cylinders. Right: Rounded ribbons are used to illustrate sheets and helices. The semi-transparent SES is shown for reference. The ligand equilenine (red) is visualized in ball-and-stick representation.

them, were proposed by Dahl et al. [35]. TexMol by Bajaj et al. [11] implements helix ray casting by using impostor-based GPU shaders.

Abstracted representations are also used for DNA and RNA. DNA is commonly depicted by a ladder-like double helix representing the sugar backbone by a ribbon or tube and the nucleotide bases by sticks or ellipsoids. Many visualization tools feature such depictions, e.g., Chimera [33] or VMD [72]. Ellipsoids are also used as a generic abstraction shape for a variety of structural elements in diverse classes of molecules [51, 2].

Although glycoscience is an active field of research, there are only few abstracted representations tailored to carbohydrate molecules. Some simple geometric abstractions of the atomic ring structures have been developed over the last decade (e.g., [34, 126]).

4.2.2 Surface Abstractions

Molecular surface abstractions are typically based on the established molecular surface models detailed in Section 4.1.2. As explained in Section 2, biological macromolecules like proteins and DNA or RNA are composed of small molecular building blocks, namely amino acids in case of proteins and nucleotides in case of DNA or RNA. A simple abstraction of the vdW surface is, therefore, to represent these building blocks by one or more tight-fitting bounding spheres that contain the individual atoms. In case of a protein, this simplification reduces the number of spheres on average by an order of magnitude, while maintaining the general shape of the protein. Similar simplifications are also used in coarse-grained molecular simulations to reduce the complexity and computation time [170, 29]. This abstraction is for example available in the molecular visualization software VMD [72] as *Beads* representation. Since the resulting surface abstraction consists of spheres, fast GPU-based ray casting can be

used for rendering.

The convolution surfaces mentioned above can be used to obtain a smooth surface abstraction if the correct parameters are chosen. A larger kernel function in combination with a higher isovalue for the surface extraction results in a smoother surface that shows the general shape of a molecule instead of individual atoms. Such smoothed surfaces are especially useful for large molecular complexes that consist of up to several million of atoms like virus capsids [95].

Cipriano and Gleicher [28] presented a surface abstraction technique based on a triangulation of the SES. It smoothens surface parts that have low frequency and are, therefore, deemed less important while maintaining salient surface features. Textures can be used to highlight removed surface features such as bumps or indentations as well as binding sites for ligands.

Several technique that map the SES to a spherical coordinate system have been proposed, e.g., by deforming the bounding sphere of the molecule until it matches the SES, thereby creating a mapping between the SES and the sphere [131]. Using this mapping, the sphere can be colored according to chemical properties of the molecule or according to the path length of the sphere deformation to highlight the shape of the molecular surface.

4.3 Structural Level of Detail

Molecular visualization often aims to render large molecular structures and their systems in real time. However, at a certain size of molecular data it becomes even difficult to visualize simple models, like the vdW surface. Since displays are restricted in the number of pixels, in scenes with many million atoms, most of them are either not inside the view frustum, occluded, or their distance to the camera is so far that their projection is smaller than a pixel. Level of detail (LOD) strategies can be applied to handle such problems. On the one hand, LOD methods can be semantic, that is, show an abstract version of the molecular structure. These approaches are especially useful to reduce clutter. On the other hand, LOD methods are often used to enhance the rendering performance, e.g., by detecting elements in the scene that are occluded by others or by using low-detail proxies for distant objects. Many existing methods mostly present a seamless visual abstraction incorporating different levels of abstraction into one molecular model.

When focusing on the semantics of molecules, molecular systems may be visualized with various degrees of structural abstraction when different parts of the system are rendered using different representations. Van der Zwan et al. [171] described their GPU implementation for visualizing continuous transitions between vdW surface, ball-and-stick and cartoon model. In addition, they proposed aspects of spatial perception and illustrativeness (cf. Section 5).

On the other hand, there are several solutions that concentrate on the spatial arrangement of the molecule. Bajaj et al. [11] presented a biochemically sensitive LOD hierarchy for molecular representations. Their hierarchical image-based rendering also allows mapping of dynamically computed physical properties onto molecular surfaces.

Later, Lee et al. [104] introduced an algorithm for view-dependent real-time surface rendering of large-scale molecular models. Their approach combines an adaptive LOD visualization of the molecular model with a high quality rendering of the active site. It is based on the two step view-dependent method. In the pre-processing stage, the mesh representing the molecular surface is simplified and classified to different LOD. In the

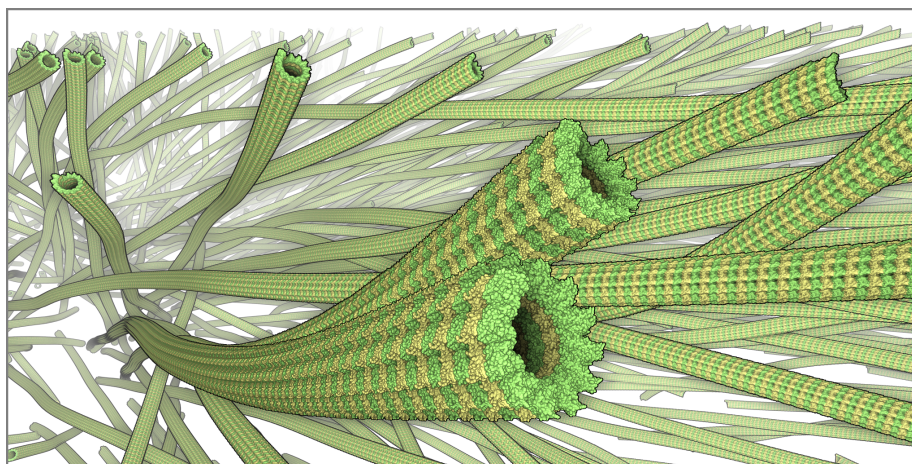


Figure 6: Microtubules reconstructed from electron tomography data and visualized as vdW surface using the approach by Lindow et al. [106] with at least 3 fps on an NVIDIA Geforce GTX 470. The data set contains 4025 microtubules with approximately 10 billion atoms.

real-time rendering stage, hierarchical LOD models which are stored in a bounding tree are constructed to increase the performance.

Convolution surfaces like the fast molecular surface extraction by Krone et al. [95] can also be used for LOD renderings. As mentioned in Section 4.2.2, this approach is able to display the structural detail on a continuous scale, ranging from atomic detail to reduced detail visual representations based on the chosen grid resolution and density kernel function. Furthermore, groups of adjacent particles can be replaced by their bounding spheres, similar to coarse-graining. If these spheres are used as an input for the convolution surface calculation, the resulting surface approximates the original shape with reduced detail.

There are a couple of methods that focus in the GPU-accelerated rendering of partly rigid structures. These methods essentially create an inverse LOD: the input data are only molecular positions from which an all-atom representation is reconstructed. In 2007, Lampe et al. [97] proposed two-level approach to visualize large, dynamic protein complexes. In the first level, each residue is reduced to a single vertex based on its rigid transformation. In the second level, the geometry shader reconstructs the atoms of the residue based on the position and orientation. The atom spheres are ray casted in the fragment shader. This approach results in a 3-fold rendering speedup; however, internal transformations of the residues are not possible. In order to minimize the data transfer to the GPU, Le Muzic et al. [100] extended this approach by storing the atom positions of a whole molecule in a texture. For each instance of a molecule, the atom positions are reconstructed using the tessellation and geometry shader. Furthermore, an LOD approach is applied, which linearly summarizes adjacent atoms into a single sphere depending on the distance to the camera. In 2012, Lindow et al. [106] presented a similar approach where the atomic data is stored in a 3D voxel grid on the GPU. During ray casting, fast ray-voxel traversal [5] is used and only spheres in the current voxel are tested for intersection. For large data sets, the rendering is much faster than direct ray casting [151] or even the two-stage culling approach by Grottel et al. [63]. Furthermore, the method exploits the fact that most biological structures, like micro-

tubules and actin filaments, consist of recurring substructures. Hence, only one grid is created for each substructure of which many instances can be rendered with different rigid transformations. This approach can be used to interactively visualize biological scenes on atomic detail bridging five orders of magnitude in length scale with billions of atoms (see Figure 6). Shortly after, Falk et al. [47] extended the technique by a hierarchical LOD to accelerate the rendering: if the projection of a grid cell is smaller than a pixel, it is not necessary to perform a ray casting with the spheres in a cell. It is only checked if the cell is empty or not. The same applies when the whole grid becomes smaller than one pixel. They also split the scene into several rendering passes. In each pass, the depth buffer of the previous pass is used for a depth test to avoid unnecessary ray casting operations. Furthermore, they presented a generalization of the approach for instances of triangulated objects. This enables the user to visualize complex models, like molecular surfaces.

Another view-dependent abstraction was proposed by Arndt et al., which is implemented in the GENOME tool [8]. They use different simple geometric abstractions to reduce detail in order to visualize the whole human genome. The simplified geometry makes it easier to identify particular components like histone proteins in an overview.

Parulek et al. [121] introduced a LOD method for fast rendering of molecular surfaces. Their method combines three molecular surface representations—SES, Gaussian convolution surface, and vdW surface—using linear interpolation (see Figure 1). The choice of the respective model is driven by an importance function that classifies the scene into three fields depending on the distance from the camera. The hierarchical abstraction incorporates a customized shading that further emphasizes the LOD. The A-buffer technique is used to improve the performance.

5 Molecular Rendering

The visualization of molecular dynamics data is often crowded and features a high visual complexity besides a high depth complexity. Advanced real-time rendering and shading methods cannot only enhance the image quality but also enhance the perception of geometric shapes and depth complexity in the scene. The main aspects related to molecular visualization are shading and various depth cues including ambient occlusion effects. In this context, the most commonly applied techniques are discussed in the following. All methods listed below have in common that they can be computed for dynamic data in real-time.

The **color** of the rendered representations is usually obtained from the type of the atoms, chains, functional units, bonds, or other derived attributes. The oldest and most simple coloring method is to assign individual colors to the chemical elements. Biochemical properties of the molecules are usually color-coded onto the atoms.

Other properties that can be mapped onto all types of molecular models using per-atom coloring include for example B-factor, flexibility, hydrophobicity, amino acid chain, or partial charge. The prevalent shading models used for **illumination** in molecular visualization are Phong [128] and Blinn-Phong [15]. However, specular highlights created with both models tend to create artifacts due to high frequencies. Grottel et al. [63] proposed a normal correction scheme to smooth out these high frequencies between adjacent normals of distant objects. This normal correction results in a more continuous lighting that creates surface-like impressions for distant molecules [63, 106].

Inspired by hand-drawn illustrations of the molecular interior of cells done by David Goodsell [57, 56], **toon shading** is often used to produce artistic or non-photorealistic

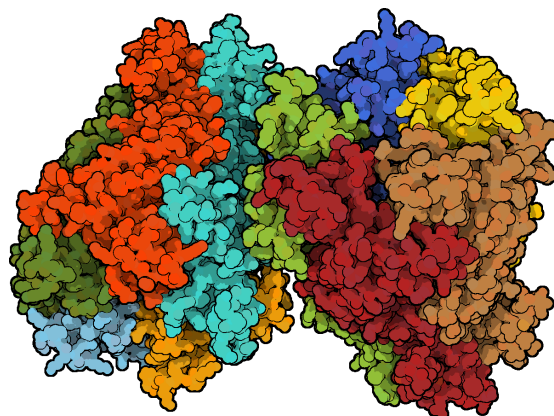


Figure 7: Non-photorealistic rendering of two proteins (PDB ID: 4A97) resembling the style used by Goodsell [56] for his Molecule-of-the-Month. Image made with MegaMol [61].

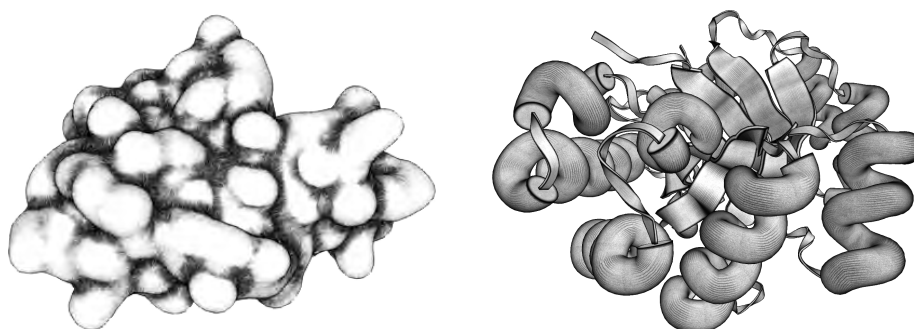


Figure 8: Illustrative line renderings of two molecules: surface structure (left, image source: [99]; PDB ID: 1OGZ) and cartoon representation (right, made with ProteinShader [178]; PDB ID: 1RWE).

renderings with a comic-like look. In Figure 7, this type of shading is applied to the protein B-Raf.

Illustrative representations using **line drawings** consisting of feature lines and hatching have a long tradition in molecular rendering. See [139] for an overview on line drawings. In particular, contour lines are widely applied in molecular visualization [165, 97, 93]. Goodsell and Olson use several types of hatching to illustrate molecular surfaces [55]. Contour lines and hatching have also been applied to yield a continuous abstraction between an atomistic model and a cartoon model of a protein [171]. The ProteinShader tool by Weber [178] offers line-based real-time illustrative rendering for cartoon representations of proteins. Lawonn et al. [99] combined feature lines and hatching to emphasize important features on molecular surfaces. The method is based on line integral convolution (LIC) on the vector field of the illumination gradient, which emphasizes salient surface regions. Figure 8 shows examples for illustrative visualizations of proteins.

Ambient Occlusion (AO) is a method based on the works of Miller [113] and Zhukov et al. [183] that mimics the transport of diffuse light between objects leading

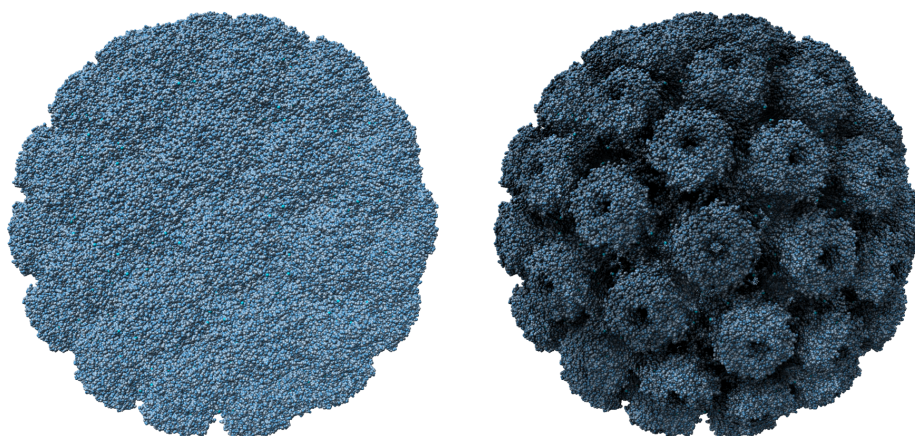


Figure 9: Rendering of a virus capsid with local illumination (left) and ambient occlusion (right). Unlike the local lighting, the ambient occlusion highlights the capsid structure clearly (made with MegaMol [61]; PDB ID: 1SVA).

to localized shadowing in creases, which can increase depth perception. AO works best for dense particle data sets, which makes it suitable for most molecular data visualizations [165]. Since AO is computationally expensive, several accelerated approaches have been developed for interactive visualization. Screen-Space AO (SSAO) is an image-space technique that approximates the effects of AO in a postprocessing step, e.g., [79]. For molecular data sets, Object-Space AO (OSAO) techniques can yield even more convincing results. OSAO considers the entire local neighborhood, unlike SSAO approaches that can only consider the visible neighborhood. Grottel et al. [62] developed an OSAO method that reaches interactive frame rates even for very large, dynamic particle data sets. In Figure 9, the differences between local illumination and OSAO are shown. Figure 2 depicts the combination of depth cueing, silhouettes, and SSAO for molecular surfaces. Note that the abovementioned list of interactive AO approaches covers only the most widely used ones for molecular visualization, as an exhaustive list of AO methods would be out of scope of this report.

Distinct object boundaries are a beneficial **depth cue** for scenes with many objects, like proteins or simulation results. Depth-dependent **silhouettes** [141] can be computed in image space in a post-processing step by detecting discontinuities in depth and adjusting line widths accordingly. A similar effect is obtained by applying **halos** extending from the object boundaries as proposed by Tarini et al. [165]. At the boundary of the object, the halo features the same depth as the object. With increasing distance from the object, the depth of the halo increases as well. A similar technique, the **depth darkening** approach by Luft et al. [110], separates distant overlapping objects visually and creates depth-dependent halos in image space. Simple fogging or depth-dependent desaturation can be used as additional depth cues.

To separate features in the foreground from the background, the **Depth of Field** (DoF) effect from photography can be used where only the objects in focus are retained sharp whereas everything else appears blurred. In molecular visualization, DoF can be used to draw the attention of the user to a specific region and is computed interactively in image space [47]. Kottraval et al. [87] recently proposed an object-space approach for DoF utilizing a coverage-based opacity estimation which can be

computed at interactive frame rates. The DoF effect can also be adjusted to highlight semantic properties [86] like single bonds or charge densities within a protein.

6 Visualization of Molecular Dynamics

As mentioned in Section 2, molecular simulation is nowadays an important source of data. Simulations can compute the individual trajectories of all atoms over a certain time frame. The resulting time-dependent data can give insight into the dynamics of the simulated molecular system on an atomistic level. Note that in this context *molecular dynamics* does not specifically refer to the results of a MD simulation, but to time-dependent molecular data that contains the dynamic behavior of the molecules.

The molecular models discussed in Section 4 can naturally be used to visualize dynamic data. They represent the instantaneous conformation of a molecule for a given snapshot and can show how it changes over time using animation. In this section, molecular visualizations are discussed that go beyond these basic models by extracting and visualizing the abovementioned dynamic behavior of the molecule. Several resources for such dynamic data exist [112, 74, 77, 14] and provide for instance short movies describing molecular functions based on their structure.

6.1 Visualization of Flexibility

Molecules are intrinsically flexible entities, yet the vast majority of visualizations represent a static structural snapshot. To account for the positional uncertainty, precisely defined atomic positions may be replaced by probability distributions to depict varying molecular conformations [133]. Representations for dynamic molecular conformations were further investigated by Schmidt-Ehrenberg et al. [147]. They developed a method to sample ball-and-stick and vdW representations onto a grid including color to depict atomic or residual properties. The conformational fuzziness thus computed is then shown using isosurface or direct volume rendering. Several programs provide “sausage” views that are similar to this method, where abstracted representations such as a protein backbone tube are modulated according to a pre-calculated flexibility parameter (see Figure 10). The width of the resulting tube highlights the flexibility. Lee and Varshney [103] depicted fuzzy atoms through multi-layered semi-transparent surfaces. More recently, Bryden et al. used glyphs to illustrate molecular flexibility calculated from normal mode analysis [19]. Their approach clusters groups of atom that exhibit a synchronized, rotational motion. The clusters are highlighted and equipped with the corresponding circular arcs that illustrate the rotation. Arrows on top of these arcs show the direction and other values like velocity, error, or nonrigid energy.

6.2 Volumetric Representations and Aggregation

Besides the specialized molecular representations discussed in Section 4, visualization methods from other fields can also be applied to depict molecular data sets. Especially vector field visualization methods can be useful for dynamic molecular data. These methods, however, require a continuous representation of the original particle data. Such a representation can be obtained by sampling points to a 3D grid. Similar to the convolution surfaces, a kernel function is often used to define the influence radius of the sampled particles. Cohen et al. [30] used volumetric maps to study accessibility. Scharnowski et al. [145] sampled dipole moments derived from the atomic positions

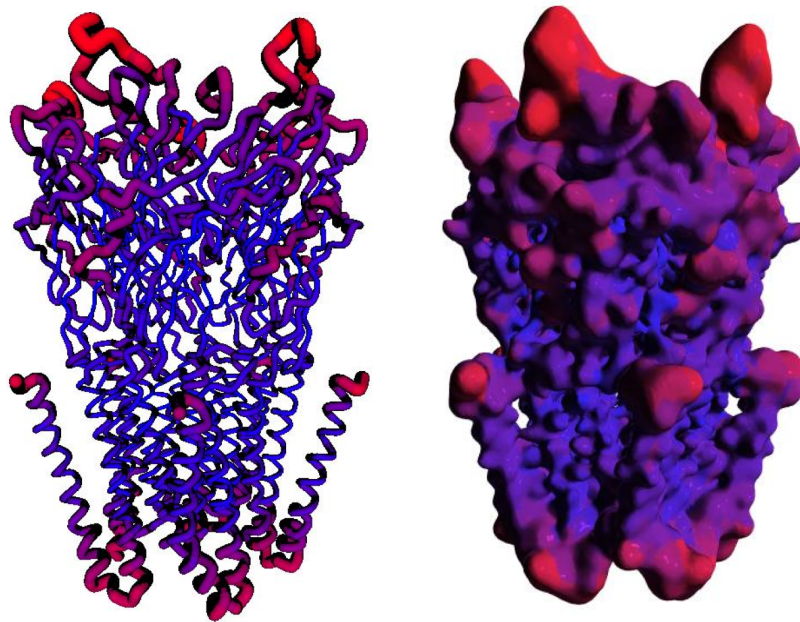


Figure 10: Representation of the backbone flexibility of the GLIC ion channel protein (PDB ID: 4HFI) by a tube of varying radius (left) or a flexibility isosurface (right). Flexible regions (red) occupy more space than well-defined rigid parts of the molecule (blue). Image generated with UnityMol [111].

to a grid and subsequently used the curl operator to separate similar regions in the resulting vector field. They rendered isosurfaces around these consistent regions. The isosurfaces were textured using line integral convolution in order to show the directions of the dipole moments. Falk et al. [46] sampled the positions of signal proteins in whole-cell simulations to a grid in order to show the development of the signal density using direct volume rendering.

Aggregation is a commonly used concept to reduce the dimensionality of scientific data. Rozmanov et al. [138] sampled atoms with different properties to separate grids in order to obtain spatial atomic densities. They also aggregated several time steps into the grid by averaging the local property values of the atoms. The aggregated densities are also visualized using isosurfaces. Temporal aggregation of atom densities and their properties was also used by Thomaß et al. [168] to visualize the average probability of presence for the components of a mixed solvent around a hydrogel. The results are mapped to an averaged molecular surface of the hydrogel using color coding. Chavent et al. [24] aggregated the diffusional motion of lipids on a grid and visualized the diffusion using arrow glyphs and streamlines. A similar approach was applied by Ertl et al. [44] to analyze the motion of ions around DNA in a nanopore. Due to the repetitive nature of the DNA and the periodic boundary conditions, they not only used temporal but also spatial aggregation of the ion densities and velocity vectors. Ertl et al. combined a multitude of visualization techniques for the analysis of the data (pathlines, isosurface, LIC, glyphs). A key point for most of the temporal aggregation methods is that the center of mass does not change significantly during the time frame of interest. Depending on the simulation, this might be given implicitly (e.g., [44]). In other cases,

a central molecule that moves freely during the simulation has to be aligned onto a given reference frame. For molecular data, alignment by RMSD minimization [78] is commonly used to superimpose the time steps.

6.3 Interactive Visualization and Haptic Manipulation of Steered Simulations

Visualization is an essential element of interactive simulations. As the visualization has to be interactive for the user to be able to steer the simulation properly, simulation performance typically is the main limiting factor. Interactivity has been a target for molecular graphics since the 1960s [48]. At that time, interaction meant essentially controlling camera movement. The element of active manipulation was added later on, first by a specialized energy minimization approach, starting with 20 to 80 residues systems, and eventually leaving out electrostatic interactions [163]. MDScope did interactive visualization and steering for MD simulations with full electrostatics up to a few hundred residues, and raised the issue of timescale limitations [119]. Intuitive haptic exploration using specialized hardware was implemented [158] and applied to a 4000 atom membrane channel. The performance requirement for haptic rendering is even more stringent than for graphics rendering, as it imposes refresh rates of about 1000 Hz. The state-of-the-art methods for atomistic molecular visualization detailed in Section 4 are able to handle dynamic data in real time. Thus, they can be used for visualizing interactive simulations. Such interactive experiments are facilitated by visual manipulation guides discussed by Kreylos et al. [90]. Nowadays, with cheaper hardware, better graphics cards, and faster computers, haptic steering has become very attractive [159] and can be applied even to systems comprising more than one million atoms [41].

6.4 Visualization of Molecular Reactions

Understanding molecular interactions in living organisms is essential to understand their physiology and is often a basis for drug design in pharmaceutical research. Modeling of coupled molecular reactions is, thus, one of the research foci in systems biology. The most widely used tools include *CellDesigner* [50], *VCell* [116], *TinkerCell* [21], *BioNetCAD* [134], *Rulebender* [154], and *NetworkViewer* [27]. Besides visualizing the quantitative change of reactants in time-intensity curve plots, these frameworks offer various forms of network visualization. These range from closely following the Systems Biology Graphical Notation [101] to illustrative textbook-like depictions of the modeled processes. However, the visualization of kinetic models primarily focuses on relational and quantitative aspects, actual behavior of involved reactants is not communicated.

Falk et al. [45] propose several methods to visually emphasize interesting aspects of particle-based cellular simulations like particle trajectories. *MCell* simulations can also be visually inspected by *CellBlender* [13], which is a plug-in for *Blender*, a 3D modeling tool. The visualization module eases generation of *MCell* models and provides direct visualization of the resulting simulation, where the molecules, represented as glyphs, are embedded into 3D meshes of cellular structures. *ZigCell3D* [37] is another system for designing and visualizing cellular models. It offers a visualization on the atomistic level while visually highlighting reactions between particles. Since such particle simulations are typically very crowded, interactions might still be missed. Thus, Le Muzic et al. [100] proposed a technique to visually represent a particle-based

system with an underlying quantitative simulation. This simulation is steered by the visualization so that reactions happen in front of the user to convey the spatial aspects of the reaction chain. Tek et al. [166] provided an environment to model and visualize protein-protein interactions. Visual cues can be complemented by multi-modal audio and haptic feedback, thus ‘rendering’ interactions calculated from live molecular simulations on multiple sensory channels.

Particle-based models have also been employed in visualization of polymerization where reactions add building blocks onto existing polymer [65]. Kolesár et al. [85] use a multiscale particle model for illustrating polymerization where the system parameters can be tweaked interactively. Thus, the user receives an instantaneous visual feedback on the growth process of the polymer.

There are a number of methods and tools to visualize covalent and non-covalent bonds [67], weak interactions [75, 32], and molecular orbitals [160] as well as related electron densities [69]. These topics, however, are out of the scope of this report.

7 Molecular Visualization Systems

In this section, our aim is not to provide the readers with an exhaustive list of existing systems for molecular visualization, as such lists are emerging quite often in the literature. We rather present the most commonly used and robust systems incorporating most of the techniques presented above.

In the last decades, many tools and systems for molecular visualization have emerged. Some of them were designed for a specific purpose and their development was stopped. On the other hand, there are several very successful and robust systems that are commonly used by domain experts in their research and publications. We decided to categorize the existing systems to three groups: freely available complex systems integrating some of the state-of-the-art methods, open-source prototype tools focused on efficient algorithms and extendability, and commercial systems. This section is structured with respect to this categorization.

The first category contains robust and popular tools, such as PyMol [39], VMD [72], Chimera [127], YASARA View [91], or CAVAR Analyst [88]. These systems are freely available for non-commercial purposes and, hence, widely used by the scientific community. Some of these systems also gain from the user community that contributes by adding own plug-ins. Most of the systems support all basic representations of molecular models discussed in Section 4. Many tools additionally provide means to equip the traditional molecular models with additional information about various physico-chemical properties and relationships in the molecular system (e.g., atomic densities, molecular orbitals, polarization, or electrostatic potentials and fields). Their proper visual representation can provide important insight into bonding and other relationships. Tools like VMD, Chimera, and PyMol furthermore enable users to load field data stored on regular grids, which can then be visualized by mesh extraction, iso-contours, or volume rendering. They also offer field line visualizations, which can be useful for electrostatics data. There is also a variety of specialized stand-alone tools for molecular visualization of such physico-chemical properties, such as Molden package [144], Molekel [130], Gabedit [4], or GaussView [40].

The second group of systems is formed by single-purpose or prototype tools, which are also freely available and most of them are open-source as well. The greatest advantage of such systems is that they focus on very efficient implementation with respect to latest advances in molecular visualization and rendering. One example is the *Qute-*

Mol tool by Tarini et al. [165], which was created to demonstrate the benefits of edge cueing and ambient occlusion. Another such tool is *ProteinShader* that showcases the illustrative cartoon rendering developed by Weber [178]. Other tools are released in the form of a prototype, sometimes as an open-source project that allows other developers to contribute. *MegaMol* by Grottel et al. [61] is an open-source rapid prototyping framework that is tailored to molecular visualization. In order to enable the development of novel, efficient visualization methods, it is designed as a thin supporting layer on top of the OpenGL API. The modular framework allows developers to add extensions by implementing plugins. The underlying core library supports the developer with basic functionality but does not restrict in terms of data structures or technologies, which is the case for some special-purpose tools. Many of the aforementioned techniques were implemented using the MegaMol framework, e.g., GPU-based cartoon models [92], molecular surfaces [93, 94, 95], and accelerated rendering and shading methods [63, 62]. *UnityMol* [111], another open-source prototype tool, was initially designed as a proof of concept to evaluate whether a game engine might enable domain scientists to easily develop and prototype novel visualizations. It was shown that a molecular viewer with original features such as animated field lines, lit spheres lighting, HyperBalls shaders [25] and more could be implemented easily and quickly. The main drawback is limited performance due to the overhead of the game engine and the nature of molecular objects. Recently, UnityMol has been extended to prototype visualizations of carbohydrate molecules [126] and act as interface for interactive molecular simulations.

The third category of systems is formed by commercial solutions like *MolSoft ICM-Pro* [1] or *Amira* [156]. They also incorporate some of the abovementioned state-of-the-art techniques. Amira, for example, provides all the classical representations like ball-and-stick, space-filling and secondary structure representations. Furthermore, molecular surfaces like vdW surface, SAS, SES, and MSS can be rendered using state-of-the-art GPU-based ray casting [108]. Amira also provides alignment and grid-based sampling tools to effectively visualize the flexibility of molecules using iso-surfaces or volume rendering. In general, however, it is often difficult to assess the commercial tools technically due to their closed source.

8 Conclusion and Future Challenges

Molecular biology is a very diverse field, which implies that the molecular visualization is diverse as well. Therefore, a plethora of different representations—each of them having its particular advantages and disadvantages—have been developed using a wide variety of visualization techniques. Consequently, there is not one best representation but rather many specialized ones, each one best suited for a specific task. One very prominent trend in the recent years has been to use the GPU not only for rendering but also to accelerate the underlying computations [23]. Programmable GPUs and multi-core CPUs have been a driving factor for parallelization of the algorithms in order to interactively visualize larger and dynamic molecular data originating from molecular simulations. At the same time, modern GPUs are powerful enough to render high-quality images at interactive frame rates. This allows domain experts to visually analyze increasingly large and complex molecular data.

The constant improvements in data acquisition technology and simulation methods provide a continuous challenge for the visualization of the derived, increasingly large molecular data sets in terms of particle numbers as well as time steps. Thus, the

development of efficient visualization algorithms remains a promising direction for future work, including out-of-core methods for the visualization of very large data sets covering long time scales. Since nowadays advances in hardware development rather increase the degree of parallelism than the clock speed, pushing the limits of parallel computing is an important issue. This includes multi-core CPUs as well as GPUs and compute clusters. Since clusters are already widely used for molecular simulation, a tight coupling of simulation and visualization methods can alleviate the in situ analysis of large molecular systems.

As compute clusters and simulation methods are improved, the system sizes that can be approached using quantum mechanics simulations also increase. In the future, novel visualization methods for the depiction of quantum effects in large dynamic molecular systems will be needed.

Another emerging trend is the use of interactive ray tracing for molecular graphics, which allows the user to get publication-quality images in real time. Sample tools that offer real-time ray tracing are BallView [114], which was one of the first tools to offer a real-time ray tracing on the CPU, and the current version of VMD, which includes a GPU-accelerated ray tracing engine [161]. Recently, Knoll et al. [84] presented a parallel interactive volume ray casting of radial basis functions on CPUs.

From more general perspective, molecular visualization will have to handle with challenges related to the increased size of input data sets. This will increase the importance of visual insight methods. Probably also new visual representations of the data will be required and in consequence, a complete visual language for biomolecules will be established.

Acknowledgments: This work was supported through grants from the German Research Foundation (DFG) as part of SFB 716, the Excellence Center at Linköping and Lund in Information Technology (ELLIIT), the French Agency for Research grants ExaViz (ANR-11-MONU-003) and Dynamo (ANR-11-LABX-0011), the Vienna Science and Technology Fund (WWTF) through project VRG11-010, the OeAD ICM through project CZ 17/2015, and the PhysioIllustration research project 218023 funded by the Norwegian Research Council.

References

- [1] Ruben Abagyan, Maxim Totrov, and Dmitry Kuznetsov. ICM—A new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation. *J. Comp. Chem.*, 15(5):488–506, 1994.
- [2] E Abrahamsson and SS Plotkin. Biovec: A program for biomolecule visualization with ellipsoidal coarse-graining. *J. Mol. Graph. Modell.*, 28(2):140–145, 2009.
- [3] Stewart A Adcock and J Andrew McCammon. Molecular dynamics: survey of methods for simulating the activity of proteins. *Chem. Rev.*, 106(5):1589–1615, 2006.
- [4] Abdul-Rahman Allouche. Gabedit - A graphical user interface for computational chemistry softwares. *J. Comput. Chem.*, 32(1):174–182, 2011.
- [5] John Amanatides and Andrew Woo. A fast voxel traversal algorithm for ray tracing. In *Eurographics*, pages 3–10, 1987.
- [6] Dirk Andrae, Ingo Barth, Timm Bredtmann, Hans-Christian Hege, Jorn Manz, Falko Marquardt, and Beate Paulus. Electronic quantum fluxes during pericyclic reactions exemplified for the cope rearrangement of semibullvalene. *J. Phys. Chem. B*, 115(18):5476–5483, 2011.
- [7] Steven S. Andrews, Nathan J. Addy, Roger Brent, and Adam P. Arkin. Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Comput. Biol.*, 6(3):e1000705, 2010.

- [8] William Arndt, Thomas M. Asbury, W. Jim Zheng, Matt Mitman, and Jijun Tang. Genome3D: A viewer-model framework for integrating and visualizing multi-scale epigenomic information within a three-dimensional genome. In *BIBM Workshops*, pages 936–938, 2011.
- [9] Richard Bader. *Atoms in molecules*. Wiley & Sons, 1990.
- [10] Ivet Bahar and AJ Rader. Coarse-grained normal mode analysis in structural biology. *Curr. Opin. Struct. Biol.*, 15(5):586–592, 2005.
- [11] C. Bajaj, P. Djeu, V. Siddavanahalli, and A. Thane. Texmol: interactive visual exploration of large flexible multi-component molecular complexes. In *IEEE Visualization*, pages 243–250, 2004.
- [12] Ingo Barth, Hans-Christian Hege, Hiroshi Ikeda, Anatole Kenfack, Michael Koppitz, Jörn Manz, Falko Marquardt, and Guennadi K Paramonov. Concerted quantum effects of electronic and nuclear fluxes in molecules. *Chem. Phys. Lett.*, 481(1):118–123, 2009.
- [13] T. M. Bartol, M. Dittrich, and J. R. Faeder. MCell. In D. Jaeger and R. Jung, editors, *Encyclopedia of Computational Neuroscience*, pages 1673–1676, New York, 2015. Springer.
- [14] Drew Berry. Molecular animation of cell death mediated by the fas pathway. *Science Signaling*, 2007(380), 2007.
- [15] James F. Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics (Proc. SIGGRAPH 1977)*, 11(2):192–198, 1977.
- [16] J.F. Blinn. A generalization of algebraic surface drawing. *ACM TOG*, 1:235–256, 1982.
- [17] Max Born. Zur Quantenmechanik der Stoßvorgänge. *Zeitschr. Phys.*, 37(12):863–867, 1926.
- [18] Siegmund Brandt and Hans Dieter Dahmen. *The picture book of quantum mechanics*. Springer, 2012.
- [19] A. Bryden, G.N. Phillips, and M. Gleicher. Automated illustration of molecular flexibility. *IEEE Trans. Vis. Comput. Graphics*, 18(1):132–145, 2012.
- [20] Tolga Can, Chao-I Chen, and Yuan-Fang Wang. Efficient molecular surface generation using level-set methods. *J. Mol. Graph. Modell.*, 31(4):442–454, 2006.
- [21] Deepak Chandran, FrankT Bergmann, and HerbertM Sauro. TinkerCell: modular CAD tool for synthetic biology. *J. Biol. Eng.*, 3(1):19–36, 2009.
- [22] Mathieu Chavent, Bruno Levy, and Bernard Maigret. MetaMol: High quality visualization of molecular skin surface. *J. Mol. Graph. Modell.*, 27(2):1391–1398, 2008.
- [23] Matthieu Chavent, Bruno Levy, Michael Krone, Katrin Bidmon, Jean-Philippe Nomine, Thomas Ertl, and Marc Baaden. GPU-powered tools boost molecular visualization. *Brief. Bioinform.*, 12(6):689–701, 2011.
- [24] Matthieu Chavent, Tyler Reddy, Joseph Goose, Anna Caroline E. Dahl, John E. Stone, Bruno Jobard, and Mark S. P. Sansom. Methodologies for the analysis of instantaneous lipid diffusion in MD simulations of large membrane systems. *Faraday Discuss.*, 169:455–475, 2014.
- [25] Matthieu Chavent, Antoine Vanel, Alex Tek, Bruno Levy, Sophie Robert, Bruno Raffin, and Marc Baaden. GPU-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids. *J. Comput. Chem.*, 32(13):2924–2935, 2011.
- [26] Ho-Lun Cheng and Xinwei Shi. Quality mesh generation for molecular skin surfaces using restricted union of balls. *Comput. Geom.*, 42(3):196–206, 2009.
- [27] Hsueh-Chien Cheng, Bastian R Angermann, Fengkai Zhang, and Martin Meier-Schellersheim. NetworkViewer: visualizing biochemical reaction networks with embedded rendering of molecular interaction rules. *BMC Syst. Biol.*, 8(70), 2014.

- [28] Gregory Cipriano and Michael Gleicher. Molecular surface abstraction. *IEEE Trans. Vis. Comput. Graphics*, 13(6):1608–1615, 2007.
- [29] Cecilia Clementi. Coarse-grained models of protein folding: toy models or predictive tools? *Curr. Opin. Struct. Biol.*, 18(1):10–15, 2008.
- [30] Jordi Cohen, Kwiseon Kim, Paul King, Michael Seibert, and Klaus Schulten. Finding gas diffusion pathways in proteins: Application to O₂ and H₂ transport in CpI [FeFe]-hydrogenase and the role of packing defects. *Structure*, 13(9):1321–1329, 2005.
- [31] M. L. Connolly. Analytical molecular surface calculation. *J. Appl. Cryst.*, 16(5):548–558, 1983.
- [32] Julia Contreras-García, Erin R. Johnson, Shahar Keinan, Robin Chaudret, Jean-Philip Piquemal, David N. Beratan, and Weitao Yang. NCIPLOT: a program for plotting noncovalent interaction regions. *J. Chem. Theory Comput.*, 7(3):625–632, 2011.
- [33] Gregory S Couch, Donna K Hendrix, and Thomas E Ferrin. Nucleic acid visualization with UCSF Chimera. *Nucleic Acids Res.*, 34(4):e29, 2006.
- [34] Simon Cross, Michelle M Kuttel, John E Stone, and James E Gain. Visualisation of cyclic and multi-branched molecules with VMD. *J. Mol. Graph. Modell.*, 28(2):131–139, 2009.
- [35] Anna Caroline E Dahl, Matthieu Chavent, and Mark S P Sansom. Bendix: intuitive helix geometry analysis and abstraction. *Bioinformatics*, 28(16):2193–2194, 2012.
- [36] John Dalton. *A New System of Chemical Philosophy, Part I and II*. Birkenstaff, London, 1808 and 1810. digital facsimile: <https://archive.org/details/newssystemofchemi01daltuoft>.
- [37] Pablo de Heras Ciechowski, Michael Klann, Robin Mange, and Heinz Koepl. From biochemical reaction networks to 3D dynamics in the cell: The ZigCell3D modeling, simulation and visualisation framework. In *IEEE Symposium on Biological Data Visualization*, pages 41–48, 2013.
- [38] Sergio Decherchi and Walter Rocchia. A general and robust ray-casting-based algorithm for triangulating surfaces at the nanoscale. *PLoS ONE*, 8(4):e59744, 2013.
- [39] W. DeLano. PyMOL: An open-source molecular graphics tool. *CCP4 Newsletter On Protein Crystallography*, 40, 2002. <http://pymol.sourceforge.net/>.
- [40] Roy Dennington, Todd Keith, and John Millam. GaussView Version 5, 2009. Semichem Inc., Shawnee Mission, KS.
- [41] M Dreher, M Piuze, A Turki, M Chavent, M Baaden, N Férey, S Limet, B Raffin, and S Robert. Interactive molecular dynamics: Scaling up to large systems. *Procedia Comput. Sci.*, 18:20–29, 2013.
- [42] Herbert Edelsbrunner. Deformable smooth surface design. *Discrete Comput. Geom.*, 21(1):87–115, 1999.
- [43] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM TOG*, 13(1):43–72, 1994.
- [44] Thomas Ertl, Michael Krone, Stefan Kesselheim, Katrin Scharnowski, Guido Reina, and Christian Holm. Visual analysis for space-time aggregation of biomolecular simulations. *Faraday Discuss.*, 169:167–178, 2014.
- [45] Martin Falk, Michael Klann, Matthias Reuss, and Thomas Ertl. Visualization of signal transduction processes in the crowded environment of the cell. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 169–176, 2009.
- [46] Martin Falk, Michael Klann, Matthias Reuss, and Thomas Ertl. 3d visualization of concentrations from stochastic agent-based signal transduction simulations. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1301–1304, 2010.

- [47] Martin Falk, Michael Krone, and Thomas Ertl. Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. *Comput. Graph. Forum*, 32(8):195–206, 2013.
- [48] Eric Francoeur. Cyrus Levinthal, the Kluge and the origins of interactive molecular graphics. *Endeavour*, 26(4):127–131, 2002.
- [49] Henry Fuchs, John Poulton, John Eyles, Trey Greer, Jack Goldfeather, David Ellsworth, Steve Molnar, Greg Turk, Brice Tebbs, and Laura Israel. Pixel-planes 5: A heterogeneous multiprocessor graphics system using processor-enhanced memories. In *SIGGRAPH '89*, pages 79–88. ACM, 1989.
- [50] A. Funahashi, M. Morohashi, H. Kitano, and N. Tanimura. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, 1(5):159–162, 2003.
- [51] A Gabriel, T Meyer, and G Germano. Molecular graphics of convex body fluids. *J. Chem. Theory Comput.*, 4:468–476, 2008.
- [52] K.D. Gibson and H.A. Scheraga. Exact calculation of the volume and surface area of fused hard-sphere molecules with unequal atomic radii. *Mol. Phys.*, 62(5):1247–1265, 1987.
- [53] Daniel T Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.
- [54] Thomas D Goddard and Thomas E Ferrin. Visualization software for molecular assemblies. *Curr. Opin. Struct. Biol.*, 17(5):587–95, 2007.
- [55] D. S.S. Goodsell and A. J. Olson. Molecular illustration in black and white. *J. Mol. Graph.*, 10:235–240, 1992.
- [56] David S. Goodsell. Molecule of the month. <http://www.rcsb.org/pdb/motm.do>, (Online, January 2015).
- [57] David S. Goodsell. *The Machinery of Life*. Springer, second edition, 2009.
- [58] David S. Goodsell. Atomistic vs. continuous representations in molecular biology. In Ray Paton and Irene Neilson, editors, *Visual Representations and Interpretations*, pages 146–155. Springer London, 1999.
- [59] J. A. Grant and B. T. Pickup. A Gaussian description of molecular shape. *J. Phys. Chem.*, 99(11):3503–3510, 1995.
- [60] Jonathan Greer and Bruce L. Bush. Macromolecular shape and surface maps by solvent exclusion. *Proc. Natl. Acad. Sci. USA*, 75:303–307, 1978.
- [61] Sebastian Grottel, Michael Krone, Christoph Müller, Guido Reina, and Thomas Ertl. MegaMol - a prototyping framework for particle-based visualization. *IEEE Trans. Vis. Comput. Graphics*, 21(2):201–214, 2015.
- [62] Sebastian Grottel, Michael Krone, Katrin Scharnowski, and Thomas Ertl. Object-space ambient occlusion for molecular dynamics. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 209–216, 2012.
- [63] Sebastian Grottel, Guido Reina, Carsten Dachsbacher, and Thomas Ertl. Coherent culling and shading for large molecular dynamics visualization. *Comput. Graph. Forum*, 29:953–962, 2010.
- [64] Sebastian Grottel, Guido Reina, and Thomas Ertl. Optimized data transfer for time-dependent, GPU-based glyphs. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 65–72, 2009.
- [65] Gerd Gruenert, Bashar Ibrahim, Thorsten Lenser, Maiko Lohel, Thomas Hinze, and Peter Dittrich. Rule-based spatial modeling with diffusing, geometrically constrained molecules. *BMC Bioinformatics*, 11(307), 2010.

- [66] Stefan Gumhold. Splatting illuminated ellipsoids with depth correction. In *Vision, Modeling, and Visualization*, pages 245–252, 2003.
- [67] D. Günther, R.A. Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. Characterizing molecular interactions in chemical systems. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2476–2485, 2014.
- [68] Andreas Halm, Lars Offen, and Dieter Fellner. Visualization of complex molecular ribbon structures at interactive rates. In *Int. Conf. on Information Visualisation*, pages 737–744, 2004.
- [69] Maciej Haranczyk and Maciej Gutowski. Visualization of molecular orbitals and the related electron densities. *J. Chem. Theory Comput.*, 4(5):689–693, 2008.
- [70] Warren J Hehre. *A Guide to Molecular Mechanics and Quantum Chemical Calculations*. Wavefunction, 2003.
- [71] Werner Heisenberg. Quantenmechanik. *Naturwiss.*, 14(45):989–994, 1926.
- [72] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD: Visual molecular dynamics. *J. Mol. Graph.*, 14:33–38, 1996.
- [73] Christiaan Huygens. *Traité de la lumière*. Pierre van der Aa, Leiden, 1690. Digital facsimile: <http://gallica.bnf.fr/ark:/12148/btv1b86069987>.
- [74] Janet H. Iwasa. Animating the model figure. *Cell*, 20(12):699–704, 2008.
- [75] Erin R. Johnson, Shahar Keinan, Paula Mori-Sánchez, Julia Contreras-García, Aron J. Cohen, and Weitao Yang. Revealing noncovalent interactions. *J. Am. Chem. Soc.*, 132(18):6498–6506, 2010.
- [76] G. T. Johnson, L. Autin, M. Al-Alusi, D. S. Goodsell, M. F. Sanner, and A. J. Olson. cellPACK: a virtual mesoscope to model and visualize structural systems biology. *Nat. Methods*, 12(1):85–91, 2015.
- [77] G. T. Johnson and S. Hertig. A guide to the visual analysis and communication of biomolecular structural data. *Nat. Rev. Mol. Cell Biol.*, 15(10):690–698, Oct 2014.
- [78] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. A*, 34(5):827–828, 1978.
- [79] Vladimir Kajalin. Screen-space ambient occlusion. In Wolfgang Engel, editor, *ShaderX⁷*, chapter 6.1, pages 413–424. Charles River Media, 2009.
- [80] Yoshihiro Kanamori, Zoltan Szego, and Tomoyuki Nishita. GPU-based fast ray casting for a large number of metaballs. *Comput. Graph. Forum*, 27(2):351–360, 2008.
- [81] Daniel Kauker, Michael Krone, Alexandros Panagiotidis, Guido Reina, and Thomas Ertl. Rendering molecular surfaces using order-independent transparency. In *Eurographics Symposium on Parallel Graphics and Visualization*, volume 13, pages 33–40, 2013.
- [82] Ioannis Kepleri. *Strena, Seu de Nive Sexangula*. Francofurti ad Moenum, 1611. Digital facsimile: <https://archive.org/details/ioanniskepleriss00kepl>.
- [83] Aaron Knoll, Maria KY Chan, Kah Chun Lau, Bin Liu, Jeffrey Greeley, Larry Curtiss, Mark Hereld, and Michael E Papka. Uncertainty classification and visualization of molecular interfaces. *Int. J. Uncertain. Quant.*, 3(2):157–169, 2013.
- [84] Aaron Knoll, Ingo Wald, Paul A. Navrátil, Anne Bowen, Khairi Reda, Michael E. Papka, and Kelly Gaither. RBF volume ray casting on multicore and manycore CPUs. *Comput. Graph. Forum*, 33(3):71–80, 2014.
- [85] Ivan Kolesár, Julius Parulek, Ivan Viola, Stefan Bruckner, Anne-Kristin Stavrum, and Helwig Hauser. Interactively illustrating polymerization using three-level model fusion. *BMC Bioinformatics*, 15(345):1–16, 2014.
- [86] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *IEEE Symposium on Information Visualization (INFOVIS)*, pages 97–104, 2001.

- [87] Sathish Kottraval, Martin Falk, Erik Sundén, and Timo Ropinski. Coverage-based opacity estimation for interactive depth of field in molecular visualization. In *IEEE Pacific Visualization Symposium (PacificVis)*, 2015. To appear.
- [88] Barbora Kozlikova, Eva Sebestova, Vilem Sustr, Jan Brezovsky, Ondrej Strnad, Lukas Daniel, David Bednar, Antonin Pavelka, Martin Manak, Martin Bezdeka, Petr Benes, Matus Kotry, Artur Gora, Jiri Damborsky, and Jiri Sochor. CAVER Analyst 1.0: Graphic tool for interactive visualization and analysis of tunnels and channels in protein structures. *Bioinformatics*, 2014.
- [89] Hendrik Anthony Kramers and Helge Holst. *The atom and the Bohr theory of its structure*. Gyldendal, London, 1923.
- [90] Oliver Kreylos, Nelson Max, Bernd Hamann, Silvia Crivell, and E Bethel. Interactive protein manipulation. In *IEEE Visualization*, pages 581–588, 2003.
- [91] Elmar Krieger and Gert Vriend. YASARA view—molecular graphics for all devices—from smartphones to workstations. *Bioinformatics*, 30(20):2981–2982, 2014.
- [92] M. Krone, K. Bidmon, and T. Ertl. GPU-based visualisation of protein secondary structure. In *Theory and Practice of Computer Graphics*, volume 8, pages 115–122, 2008.
- [93] Michael Krone, Katrin Bidmon, and Thomas Ertl. Interactive visualization of molecular surface dynamics. *IEEE Trans. Vis. Comput. Graphics*, 15(6):1391–1398, 2009.
- [94] Michael Krone, Sebastian Grottel, and Thomas Ertl. Parallel contour-buildup algorithm for the molecular surface. In *IEEE Symposium on Biological Data Visualization*, pages 17–22, 2011.
- [95] Michael Krone, John E. Stone, Thomas Ertl, and Klaus Schulten. Fast visualization of gaussian density surfaces for molecular dynamics and particle system trajectories. In *EuroVis - Short Papers*, volume 1, pages 67–71, 2012.
- [96] Nico Kruithof and Gert Vegter. Meshing skin surfaces with certified topology. *Computational Geometry*, 36(3):166–182, 2007.
- [97] Ove Daae Lampe, Ivan Viola, Nathalie Reuter, and Helwig Hauser. Two-level approach to efficient visualization of protein dynamics. *IEEE Trans. Vis. Comput. Graphics*, 13(6):1616–1623, 2007.
- [98] Patrick Laug and Houman Borouchaki. Molecular surface modeling and meshing. *Eng. Comput.*, 18(3):199–210, 2002.
- [99] Kai Lawonn, Michael Krone, Thomas Ertl, and Bernhard Preim. Line integral convolution for real-time illustration of molecular surface shape and salient regions. *Comput. Graph. Forum*, 33(3):181–190, 2014.
- [100] M. Le Muzic, J. Parulek, A.K. Stavrum, and I. Viola. Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. *Comput. Graph. Forum*, 33(3):141–150, 2014.
- [101] Nicolas Le Novère et al. The systems biology graphical notation. *Nat. Biotechnology*, 27(8):735–741, 2009.
- [102] B. Lee and F. M. Richards. The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Bio.*, 55(3):379–380, 1971.
- [103] Chang Ha Lee and Amitabh Varshney. Representing thermal vibrations and uncertainty in molecular surfaces. In *SPIE Conference on Visualization and Data Analysis*, pages 80–90, 2002.
- [104] Jun Lee, Sungjun Park, and Jee-In Kim. View-dependent rendering of large-scale molecular models using level of detail. In *Int. Conf. on Hybrid Information Technology (ICHIT)*, volume 1, pages 691–698, 2006.
- [105] Cyrus Levinthal. Molecular model-building by computer. *Sci. Am.*, 214(6):42–52, 1966.

- [106] N. Lindow, D. Baum, and H.-C. Hege. Interactive rendering of materials and biological structures on atomic and nanoscopic scale. *Comput. Graph. Forum*, 31:1325–1334, 2012.
- [107] N. Lindow, D. Baum, and H.-C. Hege. Ligand excluded surface: A new type of molecular surface. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2486–2495, 2014.
- [108] Norbert Lindow, Daniel Baum, Steffen Prohaska, and Hans-Christian Hege. Accelerated visualization of dynamic molecular surfaces. *Comput. Graph. Forum*, 29(3):943–952, 2010.
- [109] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM SIGGRAPH '87*, volume 21, pages 163–169, 1987.
- [110] Thomas Luft, Carsten Colditz, and Oliver Deussen. Image enhancement by unsharp masking the depth buffer. *ACM TOG*, 25(3):1206–1213, 2006.
- [111] Zhihan Lv, Alex Tek, Franck Da Silva, Charly Empereur-mot, Matthieu Chavent, and Marc Baaden. Game on, science - how video game technology may help biologists tackle visualization challenges. *PLoS ONE*, 8(3):e57990, 2013.
- [112] G. McGill. Molecular movies... coming to a lecture near you. *Cell*, 133(7):1127–1132, Jun 2008.
- [113] Gavin Miller. Efficient Algorithms for Local and Global Accessibility Shading. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 319–326. ACM, 1994.
- [114] Andreas Moll, Andreas Hildebrandt, Hans-Peter Lenhof, and Oliver Kohlbacher. BAL-LView: A tool for research and education in molecular modeling. *Bioinformatics*, 2005.
- [115] Steven Molnar, John Eyles, and John Poulton. Pixelflow: High-speed rendering using image composition. In *SIGGRAPH '92*, pages 231–240. ACM, 1992.
- [116] I.I. Moraru, J.C. Schaff, B.M. Slepchenko, M.L. Blinov, F. Morgan, A. Lakshminarayana, F. Gao, Y. Li, and L.M. Loew. Virtual cell modelling and simulation software environment. *IET Syst. Biol.*, 2(5):352–362, 2008.
- [117] E Moritz and J Meyer. Interactive 3D protein structure visualization using virtual reality. In *IEEE Symposium on Bioinformatics and Bioengineering*, pages 503–507, 2004.
- [118] K Namba, D L Caspar, and G J Stubbs. Computer graphics representation of levels of organization in tobacco mosaic virus structure. *Science*, 227(4688):773–776, 1985.
- [119] Mark Nelson, William Humphrey, Richard Kufrin, Attila GURSOY, Andrew Dalke, Laxmikant Kale, Robert Skeel, and Klaus Schulten. MDScope – a visual computing environment for structural biology. *Comput. Phys. Commun.*, 91(1–3):111 – 133, 1995.
- [120] Julius Parulek and Andrea Brambilla. Fast blending scheme for molecular surface representation. *IEEE Trans. Vis. Comput. Graphics*, 19(12):2653–2662, 2013.
- [121] Julius Parulek, Daniel Jönsson, Timo Ropinski, Stefan Bruckner, Anders Ynnerman, and Ivan Viola. Continuous levels-of-detail and visual abstraction for seamless molecular visualization. *Comput. Graph. Forum*, 33(6):276–287, 2014.
- [122] Julius Parulek and Ivan Viola. Implicit representation of molecular surfaces. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 217–224, 2012.
- [123] Giuseppe Patane and Michela Spagnuolo. State-of-the-art and perspectives of geometric and implicit modeling for molecular surfaces. In Walter Rocchia and Michela Spagnuolo, editors, *Computational Electrostatics for Biological Applications*, pages 157–176. Springer International Publishing, 2015.
- [124] Linus Pauling and Robert B Corey. The pleated sheet, a new layer configuration of polypeptide chains. *Proc. Natl. Acad. Sci. USA*, 37(5):251–256, 1951.
- [125] Linus Pauling, Robert B Corey, and Herman R Branson. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. USA*, 37(4):205–211, 1951.

- [126] Serge Pérez, Thibault Tubiana, Anne Imberty, and Marc Baaden. Three-dimensional representations of complex carbohydrates and polysaccharides-SweetUnityMol: A video game-based computer graphic software. *Glycobiology*, 2014.
- [127] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. UCSF Chimera - a visualization system for exploratory research and analysis. *J. Comput. Chem.*, 25(13):1605–1612, 2004.
- [128] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.
- [129] Steven J. Plimpton and Alex Slepoy. Microbial cell modeling via reacting diffusive particles. *J. Phys.: Conference Series*, 16:305–309, 2005.
- [130] Stefan Portmann and Hans Peter Luthi. MOLEKEL: An interactive molecular graphics tool. *CHIMIA International Journal for Chemistry*, 54(12):766–770, 2000.
- [131] Nicolay Postarnakevich and Rahul Singh. Global-to-local representation and visualization of molecular surfaces using deformable models. In *ACM Symposium on Applied Computing*, pages 782–787, 2009.
- [132] G. Reina and T. Ertl. Hardware-accelerated glyphs for mono- and dipoles in molecular dynamics visualization. In *EG/IEEE VGTC Symposium on Visualization*, pages 177–182, 2005.
- [133] Penny Rheingans and Shrikant Joshi. Visualization of molecules with positional uncertainty. In *Data Visualization '99*, pages 299–306, 1999.
- [134] Stéphanie Rialle, Liza Felicori, Camila Dias-Lopes, Sabine Pérès, Sanaâ El Atia, Alain R. Thierry, Patrick Amar, and Franck Molina. BioNetCAD: design, simulation and experimental validation of synthetic biochemical networks. *Bioinformatics*, 26(18):2298–2304, 2010.
- [135] F. M. Richards. Areas, volumes, packing, and protein structure. *Ann. Rev. Biophys. Bio.*, 6(1):151–176, 1977.
- [136] J Richardson. The anatomy and taxonomy of protein structure. *Advances in protein chemistry*, 34:167–339, 1981.
- [137] E. Roberts, J. E. Stone, and Z. Luthey-Schulten. Lattice Microbes: high-performance stochastic simulation method for the reaction-diffusion master equation. *J. Comput. Chem.*, 34(3):245–255, Jan 2013.
- [138] Dmitri Rozmanov, Svetlana Baoukina, and D. Peter Tieleman. Density based visualization for molecular simulation. *Faraday Discuss.*, 169(0):225–243, 2014.
- [139] S. Rusinkiewicz, F. Cole, D. DeCarlo, and A. Finkelstein. Line drawings from 3D models. In *ACM SIGGRAPH 2008 Classes*, volume 39, pages 1–356, 2008.
- [140] Joonghyun Ryu, Youngsong Cho, and Deok-Soo Kim. Triangulation of molecular surfaces. *Comput. Aided Des.*, 41(6):463–478, 2009.
- [141] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-D shapes. *Computer Graphics (Proc. SIGGRAPH 1990)*, 24(4):197–206, 1990.
- [142] M. F. Sanner and A. J. Olson. Real time surface reconstruction for moving molecular fragments. In *Pacific Symposium on Biocomputing*, pages 385–396, 1997.
- [143] Michel F. Sanner, Arthur J. Olson, and Jean-Claude Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.
- [144] G. Schaftenaar and J.H. Noordik. Molden: a pre- and post-processing program for molecular and electronic structures. *J. Comput. Aided Mol. Des.*, 14(2):123–134, 2000.
- [145] Katrin Scharnowski, Michael Krone, Filip Sadlo, Philipp Beck, Johannes Roth, Hans-Rainer Trebin, and Thomas Ertl. 2012 IEEE Visualization Contest Winner: Visualizing polarization domains in barium titanate. *IEEE Comput. Graph. Appl.*, 33(5):9–17, 2013.

- [146] Tamar Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide: An Interdisciplinary Guide*, volume 21. Springer, 2010.
- [147] Johannes Schmidt-Ehrenberg, Daniel Baum, and H-C Hege. Visualizing dynamic molecular conformations. In *IEEE Visualization*, pages 235–242, 2002.
- [148] Erwin Schrödinger. Quantisierung als Eigenwertproblem - vierte Mitteilung. *Ann. Phys.*, 81(18):109–139, 1926.
- [149] David E. Shaw, J. P. Grossman, Joseph A. Bank, Brannon Batson, and et al. Anton 2: Raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In *Int. Conf. for High Performance Computing, Networking, Storage and Analysis, SC'14*, pages 41–53, 2014.
- [150] Andrei Sherstyuk. Kernel functions in convolution surfaces: a comparative analysis. *The Visual Computer*, 15(4):171–182, 1999.
- [151] C. Sigg, T. Weyrich, M. Botsch, and M. Gross. GPU-based ray-casting of quadratic surfaces. In *EG Symposium on Point-Based Graphics*, pages 59–65, 2006.
- [152] Bernard Silvi and Andreas Savin. Classification of chemical bonds based on topological analysis of electron localization functions. *Nature*, 371(6499):683–686, 1994.
- [153] Berend Smit and Daan Frenkel. *Understanding molecular simulation: from algorithms to applications*. Elsevier Science & Technology, 2002.
- [154] A.M. Smith, W. Xu, Y. Sun, J.R. Faeder, and G.E. Marai. RuleBender: Integrated visualization for biochemical rule-based modeling. In *IEEE Symposium on Biological Data Visualization*, pages 1–8, 2011.
- [155] Deane Kingsley Smith. Bibliography on molecular and crystal structure models. Technical report, US National Bureau of Standards, 1960.
- [156] Detlev Stalling, Malte Westerhoff, and Hans-Christian Hege. Amira: a highly interactive system for visual data analysis. *The Visualization Handbook*, page 749, 2005.
- [157] Joel R. Stiles and Thomas M. Bartol. *Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology Using MCell*, chapter 4. CRC Press, 2001.
- [158] John E. Stone, Justin Gullingsrud, Klaus Schulten, and Paul Grayson. A system for interactive molecular dynamics simulation. In *ACM Symposium on Interactive 3D Graphics*, pages 191–194, 2001.
- [159] John E Stone, Axel Kohlmeyer, Kirby L Vandivort, and Klaus Schulten. Immersive molecular visualization and interactive modeling with commodity hardware. In *Advances in Visual Computing*, volume 6454 of *Lecture Notes in Computer Science*, pages 382–393. Springer, 2010.
- [160] John E. Stone, Jan Saam, David J. Hardy, Kirby L. Vandivort, Wen-mei W. Hwu, and Klaus Schulten. High performance computation and interactive display of molecular orbitals on GPUs and multi-core CPUs. In *Workshop on General Purpose Processing on Graphics Processing Units*, pages 9–18, 2009.
- [161] John E. Stone, Kirby L. Vandivort, and Klaus Schulten. GPU-accelerated molecular visualization on petascale supercomputing platforms. In *Proc. 8th Int. Workshop on Ultrascale Visualization, UltraVis '13*, pages 6:1–6:8. ACM, 2013.
- [162] Lubert Stryer. *Biochemistry*. W.H. Freeman and Company, New York, 1995.
- [163] M C Surlles, J S Richardson, D C Richardson, and F P Brooks. Sculpting proteins interactively: continual energy minimization embedded in a graphical modeling system. *Protein Sci.*, 3(2):198–210, 1994.
- [164] Laszlo Szecsi and David Illes. Real-time metaball ray casting with fragment lists. In *EG 2012 Short Papers*, pages 93–96, 2012.

- [165] Marco Tarini, Paolo Cignoni, and Claudio Montani. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Trans. Vis. Comput. Graphics*, 12(5):1237–1244, 2006.
- [166] Alex Tek, Matthieu Chavent, Marc Baaden, Olivier Delalande, Patrick Bourdot, and Nicolas Ferey. Advances in human-protein interaction—interactive and immersive molecular simulations. In Weibo Cai and Hao Hong, editors, *Protein-Protein Interactions—Computational and Experimental Tools*. InTech, 2012. Chapter 2.
- [167] Bernd Thaller. *Advanced visual quantum mechanics*. Springer, 2005.
- [168] Bertram Thomaß, Jonathan Walter, Michael Krone, Hans Hasse, and Thomas Ertl. Interactive exploration of polymer-solvent interactions. In *Vision, Modeling and Visualization*, volume 16, pages 301–308, 2011.
- [169] M. Totrov and R. Abagyan. The contour-buildup algorithm to calculate the analytical molecular surface. *J. Struct. Biol.*, 116(1):138–143, 1996.
- [170] Valentina Tozzini. Coarse-grained models for proteins. *Curr. Opin. Struct. Biol.*, 15(2):144–150, 2005.
- [171] Matthew van der Zwan, Wouter Lueks, Henk Bekker, and Tobias Isenberg. Illustrative molecular visualization with continuous abstraction. *Comput. Graph. Forum*, 30(3):683–690, 2011.
- [172] Marin van Heel, Brent Gowen, Rishi Matadeen, Elena V Orlova, Robert Finn, Tillmann Pape, Dana Cohen, Holger Stark, Ralf Schmidt, Michael Schatz, et al. Single-particle electron cryo-microscopy: towards atomic resolution. *Q. Rev. Biophys.*, 33(04):307–369, 2000.
- [173] Amitabh Varshney, Frederick P. Brooks, and William V. Wright. Linearly scalable computation of smooth molecular surfaces. *IEEE Comput. Graph. Appl.*, 14(5):19–25, 1994.
- [174] L. Velho, L. H. de Figueiredo, and J. Anthony Gomes. *Implicit Objects in Computer Graphics*. Springer, New York, first edition, 1998.
- [175] Johannes Diderik Waals. *Over de continuïteit van den gas-en vloeistoestand*. PhD thesis, Univ. Leiden, 1873.
- [176] M Wahle and S Birmanns. GPU-accelerated visualization of protein dynamics in ribbon mode. In *SPIE Conference on Visualization and Data Analysis*, pages 786–805, 2011.
- [177] JD Watson and FHC Crick. A structure for deoxyribose nucleic acid. *Nature*, 421(6921):397–398, 1953.
- [178] J. R. Weber. ProteinShader: illustrative rendering of macromolecules. *BMC Struct. Biol.*, 9(19), 2009.
- [179] Michael Mark Woolfson. *An introduction to X-ray crystallography*. Cambridge University Press, 1997.
- [180] Kurt Wüthrich. *NMR of proteins and nucleic acids*. John Wiley & Sons, New York, 1986.
- [181] Zeyun Yu. A list-based method for fast generation of molecular surfaces. In *Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, volume 31, pages 5909–5912, 2009.
- [182] Wenqi Zhao, Guoliang Xu, and Chandrajit Bajaj. An algebraic spline model of molecular surfaces. In *Symposium on Solid and Physical Modeling (SPM)*, pages 297–302, 2007.
- [183] S. Zhukov, A. Iones, and G. Kronin. An ambient light illumination model. In *Eurographics Workshop on Rendering*, pages 45–56, 1998.