Visualization of Biomolecular Structures: State of the Art Revisited

B. Kozlíková1,∗, M. Krone2,∗, M. Falk3, N. Lindow4, M. Baaden5, D. Baum4, I. Viola6,7, J. Parulek6 and H.-C. Hege4

1Faculty of Informatics, Masaryk University, Czech Republic
kozlikova@fi.muni.cz
2Visualization Research Center, University of Stuttgart, Germany
Michael.Krone@visus.uni-stuttgart.de
3Scientific Visualization Group, Linköping University, Sweden
martin.falk@liu.se
4Department of Visual Data Analysis, ZIB, Germany
{norb.lindow, baum, hege}@zib.de
5Laboratoire de Biochimie Théorique, UPR 9080 CNRS, France
baaden@smplinux.de
6Department of Informatics, University of Bergen, Norway
viola@cg.tuwien.ac.at, julius.parulek@uib.no
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.

Keywords: molecular visualization, taxonomy, bioinformatics visualization

ACM CCS: Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling—Curve, surface, solid, and object representations

1. Introduction
Interactive molecular visualization is one of the oldest branches of data visualization [Fra02], 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. This paper is an extended version of our previous survey [KKL*15] and includes newer work that was not available at that time as well as references that are historically interesting and provided the foundations for the current state of the art.

∗These authors contributed equally.

© 2016 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
either strong (covalent, ionic, and metallic bonds) or weak (dipole-dipole interactions and hydrogen bonds). Strong bonds hold sets of atoms together, forming tight entities like molecules, ionic salts, and metals. A molecule thus is a structure composed of nuclei, defining the atom positions, and core electrons (inner electron shells); the 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, 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 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.

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) [Kep11] and Huygens (1690) [Huy90], centuries before 1808, when Dalton published the modern, but still pre-quantum formulation of atomic theory [Dal10]. In these groundbreaking works, atomic arrangements were illustrated, displaying atoms as spheres. Van der Waals [vdW73] 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. Physical models of molecules, both static and dynamic, have been used for visualization purposes [Smi60].

With the emergence of increasingly elaborate atomic models by Thompson, Rutherford, Bohr and Sommerfeld in the early 20th century, more detailed visualizations became necessary, culminating in detailed depictions of complex atoms showing the elliptic orbits of electrons in the Bohr–Sommerfeld model [KH23]. 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 [Hei26]. Instead, according to Born [Bor26], an electron’s position is described by a probability distribution, given by the absolute square of Schrödinger’s complex wave function $\Psi$. The evolution of $\Psi$ for a system of $N$ quantum particles, described by the time-dependent Schrödinger equation [Sch26], happens not in real three-dimensional space, but in 3N-dimensional space of all particles’ coordinates. This poses a further challenge to visualization. Regarding visualization of fully quantum physical systems only very limited work is available; examples are [Tha05, BD12].

Fortunately, research revealed that molecular systems can be classically described 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; instead 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) [LR71]. 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 types of biological systems require quantum mechanical considerations for a detailed understanding. Examples of biological and medical relevance are enzymatic reactions or photosynthesis. See, e.g., [ADP08, AKM14] for popular-science presentations of the emerging ‘quantum biology’. This opens up a new field of research in molecular visualization, on which we will report only very briefly.

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 (Sections 4 to 6). The report is concluded by a brief overview of molecular visualization tools (Section 7) and anticipated future challenges (Section 8).

Please note that this report does not discuss methods to extract and visualize cavities in biomolecules (e.g. tunnels, pores or channels). Interested readers are referred to our recent report on this topic [KKL*16].

2. Molecular Data

This section introduces the input data, mostly formed by biomolecules, along with their composition and basic properties. Moreover, the most common sources of molecular structures and molecular dynamics are discussed.

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 [Str95]. 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 types of biomolecules are briefly introduced.

The building blocks of nucleic acids are nucleotides consisting of a nucleobase, a sugar, and a phosphate group. The main difference between deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) is the sugar: deoxyribose in DNA and ribose in RNA. Additionally, 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 [WC53]. In contrast, RNA is single-stranded and typically forms very complex structures. DNA stores the genetic code including the information about the composition of proteins.

Proteins are macromolecules consisting of one or more chains of amino acids. Different proteins have diverse functions like replica-tion of DNA, catalyzing chemical reactions, or transport of other molecules. The amino acids forming the protein are connected via peptide bonds. This chain is called the protein’s primary structure. The amino acid chain folds into an energetically favourable 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 α-helix [PCB51] and β-sheet [PC51], which are connected by loops and unstructured parts called random coil. The correct folding of the chain is important for the function of most proteins. The 3D 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. In the visualization literature, the term secondary structure sometimes is used synonymously for tertiary and quaternary one, see, e.g., [WB11].

Lipids and lipid membranes are ubiquitous in biological systems as they delineate the compartments of the cell, control entry and transport, and harbour 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 necessary for life [Goo09]; actually they are 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 the following three techniques: X-ray crystallography [Woo97], which potentially leads to the highest resolution data when crystals can be obtained; nuclear magnetic resonance spectroscopy (NMR) [Wit86] determining structural ensembles rather than a single structure; cryo-electron microscopy (cryo-EM) [vHGM’00] allowing the determination of large structures, but requiring an image-based reconstruction with limited resolution. Visualization can aid the structure determination process as a complement to image processing and classification algorithms.

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 environmental parameters and the interaction with other molecules. Modern GPU-accelerated quantum mechanics simulations can still only simulate small proteins [KLUM12]. Thus, for larger systems, the most frequently used methods are Monte Carlo (MC) sampling and molecular dynamics (MD) simulations. An introduction to these methods can be found in the textbook by Frenkel and Smit [SF02]. Both methods usually do not model quantum mechanical effects explicitly but incorporate such effects only through classical molecular force fields. 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), it is computationally very expensive to simulate the system for relevant time intervals of milliseconds or even seconds. Although Shaw et al. [GB*14] demonstrated that it is possible to run 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. Here, groups of atoms instead of single atoms are considered as the smallest unit. Depending on the molecular systems, several types of coarse-grained models can be adopted (see, e.g., [Cle08]). Recently, Krieger and Vriend [KV15] introduced a set of algorithms to improve the performance of MD simulations. If a simulation process is mainly controlled by diffusion, Brownian Dynamics is often used as a complementary approach to MD [AM06].

The results of molecular modelling 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 centre 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 [BR05]. 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, while trading accuracy.

Recently, Johnson et al. developed a semi-automatic modelling tool called cellPack [JAAA*15] that computes a packing of molecules to form comprehensive models of very complex molecular systems up to mesoscopic length scales.

Another data source are biochemical reaction models, which can be categorized roughly as kinetic models and particle-based ones. 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 [SB01],
ChemCell [PS05], and Smoldyn [AABA10], covering membrane interaction, diffusion, and 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 [Gil07]. As recently shown [RSLS13], the chemical master equation and the reaction-diffusion master equation, both underlying stochastic simulations, can be efficiently sampled on GPUs, speeding up the computation by up to two orders of magnitude.

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 types of visualization can be subdivided into showing static geometry (left side) or depicting an animation (right side). Visualizing static geometry results in a still image. Such an image can nonetheless show dynamic properties or attributes derived from these. The animation on the other hand focuses on real-time playback to further emphasize features related to dynamics. 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. Furthermore, additional bioinformatics data like phylogenetic trees and other biomolecular information can be included as well.

The coloured 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), visualization of molecular reactions (violet, Section 6.4), and visualization of quantum effects (blue, Section 6.5). 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.

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 illustrated taxonomy in Figure 1. Large molecular systems are often depicted using level of detail visualizations (Section 4.3), which include continuous representations as defined by Goodsell [Goo99] that simplify the atomic details.

4.1. Atomistic models

Atomistic models directly depict the atoms of a molecule. The atomic structure plays an essential role in determining molecular properties. Atomistic representations model discrete entities and can be used in molecular systems consisting of up to millions of atoms. They can be classified into models that focus on atomic bonds and surface models that show the interface between a molecule and its environment.

In traditional interactive molecular graphics, molecular models are 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 as presented by Gumhold [Gum03] to render ellipsoids became more efficient. The general idea is to render a projection of a primitive that encloses the 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 [RE05] used a combined ray casting of spheres and cylinders to visualize mono- and dipoles in MD data. Sigg et al. [SWBG06] 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 surfaces in real-time with pixel-perfect quality for any zoom level.

4.1.1. Bond-centric models

Visualizing chemical bonds between atoms helps to understand and to 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, GPU-based ray casting is much more efficient and achieves higher visual quality than triangle-based rendering for such implicit objects. However, most modern techniques for bond representation are descendants of techniques and software tools that came out in the late 1980s and early 1990s [FPE*89, MEP92].

Chavent et al. [CVT*11] 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 Models and Van der Waals Surfaces.** 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* [Ric77] is a space-filling model where the radius of the atom spheres is proportional to the van der Waals radius. This surface shows the molecular volume, that is, it illustrates the spatial volume the molecule occupies. The vdW surface is the basis of most other molecular surface representations (Figure 4). In 1995, Sayle and Milner-White presented the molecular graphics tool *RasMol* [SMW95], which was one of the first tools supporting fast visualization of the vdW representation and exploited CPUs for rendering. Nowadays, GPU-based ray casting of the vdW spheres is the fastest way to visualize the vdW surface of several million of atoms [GRE09]. Recently, further techniques were proposed to handle even larger data sets (see Section 4.3).

**Solvent Accessible Surface.** Lee and Richards defined one of the first extensions to the vdW surface, the *solvent accessible surface* (SAS) [LR71]. The idea 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 centre of the probe while rolling over the vdW surface (see 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 centre 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 [Ric77] defined the first smooth molecular surface (see Figure 4) based on the idea of the SAS. Instead of taking the centre of the probe that rolls over the atoms, he suggested to use the boundary of the spherical 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 [GB78], which is equivalent 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 term solvent excluded surface (SES). Figure 2 gives an overview of all publications concerning SES visualization. Mathematically, the SES is composed of three types of patches: Convex spherical patches occur where the probe touches exactly one atom; toroidal 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, i.e. the SES is smooth. However, the surface can contain self-intersections, also called ‘singularities’ [SOS96]. Here, the surface has sharp edges and is only \(C^0\)-continuous. Two types of singularities 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 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.

The algorithms for computing the SES fall into two categories. The first comprises all methods that compute the surface by discretizing the space \(\mathbb{R}^3\). These approaches usually compute a discrete scalar field from which an isosurface is extracted, either by triangulation via Marching Cubes [LC87] or by direct isosurface ray marching. Two of the fastest approaches in this research area were presented by Can et al. [CCW06] and Yu [Yu09]. 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 [Con83] presented the equations to compute the SES analytically and the first algorithm based on this. Varshney et al. [VBW94] proposed a parallel algorithm based on the computation of an approximate Voronoi diagram. Edelsbrunner and Mücke [EM94] introduced alpha shapes that can be used to compute the SES. Sanner et al. [SOS96] 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 [SO97]. In 2009, Krone et al. [KBE09] 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, Trotrov and Abagyan [TA96] proposed the contour-buildup (CB) algorithm. It directly computes the track of the probe on each atom surface and therefore is embarrassingly parallel. Lindow et al. [LBPH10] presented a parallel CB algorithm using OpenMP, which allowed the user to visualize dynamic molecules with up to \(10^7\) atoms on 6 core systems. Krone et al. [KGE11] parallelized the CB algorithm for GPUs, which further accelerated the SES computation and enabled the interactive visualization of dynamic molecules with up to \(10^7\) atoms. These two methods are currently the fastest analytical techniques to compute the SES.

For rendering, the SES was traditionally tessellated. Examples for very accurate tessellations are the one by Sanner et al. [SOS96] and the one by Laug and Borouchaki [LB02]. Later, Zhao et al. [ZXB07] 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. [RCK09] using subdivision surfaces. Their approach, however, is not able to handle all possible singularities.

Triangulating the SES is computationally expensive and usually takes seconds for mid-sized proteins. In 2009, Krone et al. [KBE09] thus used GPU-based ray casting to render the three types of surface patches. As mentioned above, it yields not only 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. [LBPH10] presented a slightly improved ray casting that uses the geometry shader to optimize the rasterization of primitives, which is approximately 30% faster. To optimize the ray casting performance, the parts of the convex spherical patches lying inside the SES were not clipped in these previous methods. Hence, the surface could be visualized only opaquely 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. [KKP+13]. Recently, Jurčík et al. [JPSS16] presented an improved transparent rendering of the SES based on the fast GPU-accelerated SES computation of Krone et al. [KGE11]. 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 [PV12]. They use a modified sphere tracing and directly compute the implicit description of the surface based on the local neighbourhood 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 artefacts, e.g. at singularities and patch boundaries. Details can be found in the STAR by Patane and Spagnuolo [PS15] on geometric and implicit modelling for molecular surfaces.

Decherchi and Rocchia [DR13] presented a combination of triangulation and ray casting. Their 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 managed to accelerate the triangulation of the SES, the overall speed and visual quality cannot compete with direct ray casting.
Figure 2: A graph showing the chronology and relation of publications addressing the visualization of solvent excluded surfaces.

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

Figure 4: Comparison between different molecular surfaces of the protein isomerase (PDB ID: 1OGZ). (a) vdW surface, (b) SES with probe radius 1.4 Å, (c) LES for equilenine, (d) MSS with shrink factor 0.35 and (e) Gaussian convolution surface with standard deviation equal to the atom radius. The ligand equilenine (red) is shown as (a), (e) stick, (b), (d) ball-and-stick, or (c) vdW surface. In all examples, depth cueing, screen-space ambient occlusion and silhouettes were applied.

Molecular Skin Surface. Edelsbrunner presented a new smooth surface for a finite set of input spheres, called skin surface [Ede99]. Its shape depends on a single parameter $s \in (0, 1)$, 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 4). Furthermore, it can be decomposed into patches of quadrics. However, the MSS has no biophysical background. Kruithof and Vegter [KV07] presented a tessellation approach for the MSS. Cheng and Shi [CS09] developed a triangulation algorithm that achieves a higher quality but is very time consuming. A very fast triangulation was presented by Decherchi and Rocchia [DR13] following the same strategy as their SES approach. However, it does not necessarily preserve the full surface topology. To achieve fast, high-quality visualization, Chavent et al. [CLM08] presented the first GPU-based ray casting of 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. [LBPH10] presented an accelerated computation using the same idea that Varshney et al. [VBW94] 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 Surfaces. The ligand excluded surface is a generalization of the SES (see Figure 4). It was recently proposed by Lindow et al. [LBH14]. In contrast to the SES, the LES does not approximate the ligand by a sphere but uses the full and potentially dynamic geometry defined by the ligand’s vdW surfaces. Thus, the LES shows the geometrical surface that a specific ligand can access when approaching the molecule. An analytical computation of the LES is difficult. Lindow et al. therefore proposed an algorithm to compute the surface by discretizing the possible ligand positions, orientations, and dynamics. While 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 favourable. The LES should be favoured if a more detailed view of a static molecule is needed.

**Convolution Surface Models.** Blinn [Bli82] introduced implicit modelling as an approximation of the molecular surface in 1982. He proposed the use of a Gaussian convolution kernel (see Figure 4) in order to blend atom potentials to represent the electron density function. The resulting surface is commonly known as Metaballs, blobby surfaces, or convolution surfaces [VFG98]. Such a summation-based model, however, generally lacks information of the associated solvent molecule. Therefore, Grant and Pickup [GP95] determined the parameters for the Gaussian-based model to mimic the volume and solvent accessible surface area for different solvent probe sizes.

There are several other kernels mentioned in the literature that can be used as alternative kernel functions [She99], i.e. avoiding computationally expensive exponential functions. One of the main advantages of kernel-based models is the simplicity of the 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 [PB13] proposed another implicit model with linear complexity although its definition is not purely analytical compared, e.g. to the Gaussian model. On the other hand, it resembles the SES more closely than the kernel-based approaches (Figure 5). 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. [KSN08] 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 algorithm, Szecsi and Illes [SI12] 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. [KSES12] 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

![Figure 5: Molecular surface for proliferating cell nuclear antigen (PDB ID: 4D2G) represented by an implicit model [PB13]. The images show the result for different solvent radii: 1.4 Å (left) and 2.2 Å (right).](Image)

is currently among the fastest molecular surface extraction algorithms.

### 4.2. Illustrative and abstract models

Apart from molecular models that directly depict 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. These models can also lead to sparse representations, which might be easier to understand or reduce occlusion. Abstract representation can also be useful for very large molecular complexes, for which often not the individual atoms but the overall shape is of interest.

#### 4.2.1. Representations of molecular architecture

Very early on, the conceptualization of complex macromolecular assemblies motivated 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 [MM04], e.g. using abstractions for molecular subunit structures [NCS85]. Goddard and Ferrin alternatively refer to such abstractions as multiple levels of detail that match the underlying structural hierarchy of molecular assemblies [GF07]. As our understanding of biological structures progresses, the need for new abstractions may arise such as it was the case for representing the bases of nucleic acid polymers and, more recently, carbohydrates.

In 1981, Richardson [Ric81] 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 6). One of the earliest implementations of the cartoon model was the *Ribbons* program [CB86], which was influential to subsequent work. Its successor *Ribbons 2.0* [Car91] provided interactive visualization. A current challenge is to improve the efficiency for the interactive visualization of large, dynamic proteins. This can for example be achieved by mesh-refinement techniques at the software level [HOF04] or by moving to GPUs at the hardware level. TexMol by Bajaj et al. [BDST04] implements
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. [CKSG09, PTIB14].

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. In a simple abstraction of the vDW surface one represents these building blocks by one or more tight-fitting bounding spheres that contain the individual atoms (e.g. beads representation in the molecular visualization tool VMD [HDS96]). 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 [Toz05, Cle08]. 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 correct parameter values are chosen. A larger kernel function in combination with a higher iso-value 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 consisting of up to several millions of atoms like virus capsids [KSES12].

Cipriano and Gleicher [CG07] 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 techniques that map a molecular surface mesh (typically a triangulated SES) to a spherical coordinate system have been proposed. Rahi and Sharp [RS07] developed a method that uses a parametrization based on spherical coordinates to map the triangles of a molecular surface onto a sphere. The technique of Postnarekvich and Singh [PS09] uses a force-directed approach to deform a bounding sphere until it matches the SES, thereby creating a mapping between the SES and the sphere. Using this mapping, the sphere can be coloured according to physico-chemical properties of the molecule or according to the path length of the sphere deformation to highlight the shape of the original SES. Hass and Koehl [HK14] use a conformal mapping between the molecular surface and a bounding sphere to measure how spherical the molecule is. They also propose to use their spherical representation to compare molecules.

4.3. Structural level of detail

Molecular visualization often aims to render large molecular structures and 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 atoms are either not inside the view frustum, occluded, or so distant to the camera that their projection is significantly 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; such 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. Most existing methods present a seamless visual abstraction, incorporating different levels of abstraction into one molecular model.

When focusing on the semantics, molecular systems may be visualized with various degrees of structural abstraction, i.e. different parts of the system are rendered using different representations. Van der Zwan et al. [vdZLBI11] described a GPU implementation for visualizing continuous transitions between vDW surface, ball-and-stick, and cartoon model. They also proposed methods to support spatial perception and enhance illustrative power (cf. Section 5).

On the other hand, there are several solutions that focus on the spatial arrangement of molecules. Bajaj et al. [BDST04] 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. [LPK06] introduced an algorithm for viewpoint-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 a two-step view-dependent method: In a pre-processing stage, the mesh representing the molecular surface is simplified and classified to different LODs; in a 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. [KSES12] 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. Lampe et al. [LVRH07] proposed a 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-cast in the fragment shader. An additional feature is the fish-eye distortion, which allows the user to get a better view inside the protein. This approach results in a three-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. [LMPSV14] extended this approach by storing the atom positions of a whole molecule in a texture. Each instance of the molecule is then formed just by a single vertex, where 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 contrast to Lampe et al. [LVRH07], this LOD approach is not restricted to protein data. Later on, Le Muzic et al. [LMAPV15] presented a system, cellVIEW, to interactively visualize large molecular datasets using the Unity3D game engine (see Figure 7). The exploited techniques further advanced the performance of atomistic visualization by means of a real-time LOD selection technique implemented in the tessellation shader. The proposed approach allows to render datasets containing 15 billion atoms at 60 fps.

In 2012, Lindow et al. [LBH12] presented an approach similar to those of Le Muzic et al. [LMPSV14, LMAPV15], where the atomic data is stored in a 3D voxel grid on the GPU. During ray casting, a fast ray-voxel traversal is used and only spheres in the current voxel are tested for intersection. For very large data sets, the rendering is much faster than direct ray casting [RE05, SWBG06] or even the two-stage culling approach by Grottel et al. [GRDE10]. Furthermore, the method exploits the fact that most biological structures, like microtubules 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 8). Shortly after, Falk et al. [FKE13] accelerated the technique using a hierarchical LOD: if the projection of a grid cell is smaller than a pixel, it is not necessary to perform ray casting for the spheres in this cell. It is only checked if the cell is empty or not. The same applies when

![Figure 7: Illustrative visualization of a virus (HIV) generated by cellVIEW [LMAPV15].](image-url)
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. They also presented a generalization of the approach for instances of triangulated objects, which 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 [AAZ*11]. 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. [PJR*14] 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. The most commonly applied techniques in this context are discussed in the following. All methods listed below have in common that they can be computed for dynamic data in real-time.

The colour 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 colouring method is to assign individual colours to the chemical elements. Biochemical properties of the molecules are usually colour-coded onto the atoms.

Other properties that can be mapped onto all types of molecular models using per-atom colouring 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 [Pho75] and Blinn-Phong [Bli77]. However, specular highlights created with both models tend to create artefacts due to high frequencies. Grottel et al. [GRDE10] 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 [GRDE10, LBH12].

Inspired by hand-drawn illustrations of the molecular interior of cells done by David Goodsell [Goo09, Goo], toon shading is often used to produce artistic or non-photorealistic renderings with a comic-like look. In Figure 9, 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 [RCDF08] for an overview on line drawings. In particular, contour lines are widely applied in molecular visualization (see, e.g., [TCM06, LVRH07, KBE09]). Goodsell and Olson use several types of hatching to illustrate molecular surfaces [GO92]. Contour lines and hatching have also been applied to yield a continuous abstraction between an atomistic model and a cartoon model of a protein [vdZLBI11]. The ProteinShader tool by Weber [Web09] offers line-based real-time illustrative rendering for cartoon representations of proteins. Lawonn et al. [LKEP14] 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 10 shows examples for illustrative visualizations of proteins.
Ambient Occlusion (AO) is a method based on the works of Miller [Mil94] and Zhukov et al. [ZIK98] that mimics the transport of diffuse light between objects leading 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 [TCM06]. In Figure 11, the differences between local illumination and OSAO are shown. 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 post-processing step, e.g. [Kaj09]. For molecular data sets, Object-Space AO (OSAO) techniques can yield even more convincing results. OSAO considers the entire local neighbourhood, unlike SSAO approaches that can only consider the visible neighbourhood. Grottel et al. [GKSE12] developed an OSAO method that reaches interactive frame rates even for very large, dynamic particle data sets. The method uses a volumetric approximation of the local neighbourhood to store the ambient occlusion factors. Recently, this approach was extended by Staib et al. [SGG15] using a hierarchical voxel-cone tracing method improving the sampling of a full-colour AO map. Their method also works for transparent particles. Eichelbaum et al. [ESH13] presented PointAO, a SSAO method for particle rendering that focuses on retaining both global and local structural information. Wahl and Wriggers [WW15] developed a multi-scale SSAO method designed to highlight structural features of biomolecules. Hermosilla et al. [HGV16] presented an interactive method to generate halos and AO effects. Figure 4 depicts a combination of depth cueing, silhouettes, and SSAO for molecular surfaces. The abovementioned interactive AO approaches are only the most widely used ones for molecular visualization, as a comprehensive 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 [ST90] 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. [TCM06]. 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. [LCD06], 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 [FKE13]. Kottravel et al. [KFSR15] 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 [KMH01] like single bonds or charge densities within a protein.

Typically, the viewpoint and camera parameters are chosen by the user when rendering and exploring molecular scenes. The automatic choice of the best view for a particular molecule requires additional information besides the structural data to map the 3D structure onto the screen. Vazquez et al. [VFSL02] utilize the concept of viewpoint entropy and extend it to orthographic molecular views. Incorporating additional semantic information on the protein can improve the selection of an optimal camera setting [DCMP10].

Besides the rendering techniques that highlight shape and depth complexity of the data, stereoscopic rendering is widely used in molecular graphics (see, e.g., [GF07]). While stereoscopic rendering requires special hardware like head-mounted displays (HMD), 3D glasses, or auto-stereoscopic screens, the rendering part is usually relatively straightforward: for each eye, a separate image is rendered, each with the appropriate camera settings. Obviously, the rendering also requires twice the compute power. Recently, Stone et al. [SSS16] presented a remote rendering system for the Oculus Rift HMD that uses real-time ray tracing. It is noteworthy that the use of the abovementioned rendering methods to highlight shape has to be considered carefully for stereoscopic rendering since these methods are designed for monoscopic rendering and can lead to perceptual issues.

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 provide 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 represents 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 and provide for instance short movies describing molecular functions based on their structure (e.g. [Ber07, Iwa08, JH14]). These educational solutions mainly focus on the artistic appearance and use pre-rendered, non-interactive visualizations.

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 [RJ99]. Representations for dynamic molecular conformations were further investigated by Schmidt-Ehrenberg et al. [SEBH02]. They developed a method to sample ball-and-stick and vdW representations onto a grid including colour to depict atomic or residual properties. The conformational fuzziness thus computed is then shown using isosurface or direct volume rendering. MolMol [KBW96] and several other 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 12). The width of the resulting tube highlights the flexibility. Lee and Varshney [LV02] depicted thermal vibrations of atoms through multi-layered semi-transparent surfaces. Selected flexible elements such as loops or domains in proteins can be represented by voxel maps [CBES11].

Bryden et al. [BPG12] used glyphs to illustrate molecular flexibility calculated from normal mode analysis. Their approach clusters groups of atoms 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 of the rotation and other values like velocity, error, or non-rigid energy. Fioravante et al. [FSTR13] presented visualization methods that uses principal component analysis and covariance clustering to analyze motional correlations in proteins. The results of these analyses are used to enrich the 3D visualization of the protein structure, e.g. using colour or cone glyphs. Ahlstrom [ABE*13] presented a similar approach that uses network visualization to show different conformations that occur during MD simulation.

Heinrich et al. [HKOW14] presented a visual analysis application tailored to intrinsically disordered proteins. Such proteins have very flexible regions that can exhibit a wide range of three-dimensional structures depending on external factors [UD10]. The application shows a 3D visualization of an ensemble of superimposed structures as well as a parallel coordinates plot [Ins09] with per-residue statistics. This plot can be used to filter or cluster the protein structures and to find correlations between them.

Recently, Dabdoub et al. [DRSR15] presented the tool MoFlow that visualizes the dynamics of a molecule by rendering the pathlines of selected atoms of the molecular structure, e.g. backbone atoms. The atom positions between time steps are interpolated using splines. The resulting curves are coloured according to a timescale colour map allowing an easy understanding of the movements of the atoms over time. More visual cues are added through semi-transparent ribbons displaying the movement of bonds. While MoFlow allows an easy understanding of short parts of a trajectory, the visual representation might quickly get confusing for very complex movements.

6.2. Volumetric representations and aggregation

Besides the tailored representations discussed in Section 4, visualization methods developed for other application fields can also be utilized 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 raw particle data. Such representations 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. [CKK*05] used volumetric maps to study accessibility. Scharnowski et al. [SKS*13] sampled dipole moments derived from the atomic positions 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. Line integral convolution on these surfaces shows the directions of the dipole moments. Falk et al. [FKRE10] sampled the positions of signal proteins in whole-cell simulations to a grid to show the development of the signal density using direct volume rendering (Figure 13).

Aggregation is a commonly used concept to reduce the dimensionality of scientific data. Rozmanov et al. [RBT14] sampled atoms with different properties to separate grids to obtain spatial atomic densities. They also aggregated several time steps into a grid by averaging 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. [TWK*11] to visualize the average probability of presence for the components of a mixed solvent around a hydrogel. The results are colour-mapped to an averaged molecular surface of the hydrogel. An alternative representation is to use volume rendering as did Durrieu et al. [DLB08] to illustrate water occupancy around a protein averaged over a MD simulation. In specific cases, such as when a cylindrical geometry around a protein channel is observed, the dimensionality of the representation can be further reduced to map, e.g. the solvent density in 2D as in [BS03]. Chavent et al. [CRG*14] aggregated the diffusional motion of lipids on a grid and visualized the diffusion using arrow glyphs and streamlines. A similar approach was used by Ertl et al. [EKK*14] 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 velocities. They combined different visualization methods for the analysis of the data (pathlines, isosurface, LIC, glyphs). A key point for most temporal aggregation methods is that the centre of mass does not change significantly during the time frame of interest. Depending on the simulation, this might be given implicitly (e.g. [EKK*14]). Otherwise, a central molecule that moves freely during the simulation has to be aligned onto a reference frame. For molecules, alignment by RMSD minimization [Kab78] is commonly used to superimpose all frames.

Aggregation and clustering has not only been applied to grid-based molecular data. Lindow et al. [LBBH12] for example used aggregation to illustrate time-dependent channels of proteins. Their method extracts the channel state for each time step and aggregates partially open channels if they are connected in consecutive time steps to show in one static image whether the they can be traversed over time. Byška et al. [BJG*15] used an aggregated bottleneck contour (collar) to focus on a given channel constriction. Bidmon et al. [BGB*08] aggregated solvent pathlines—i.e. the trajectories of individual solvent molecules—to investigate the water movement near protein cavities.

6.3. Interactive visualization and manipulation of molecular models

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 [Fra02]. 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 [SRRB94]. MD-Scope did interactive visualization and steering for MD simulations with full electrostatics up to a few hundred residues, and raised the issue of timescale limitations [NHK*95]. Especially in the context of steered simulations, haptic feedback using specialized interaction devices becomes interesting since it can be used to convey forces. Another application area that uses methods for direct manipulation of molecular data is interactive molecular modelling. Intuitive haptic exploration using specialized hardware was implemented [SGSG01] 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 modern molecular visualization methods detailed in Section 4 are able to handle dynamic data in real time. Thus, they can be used for visualizing interactive simulations. A typical setup is depicted in Figure 14. Nowadays, even chemical reactivity can be explored interactively using QM simulations [HVB*14, LJM15]. Such interactive experiments are facilitated by visual manipulation guides discussed by Kreylos et al. [KMH*03]. They developed the molecular-modelling tool ProteinShop that uses inverse kinematics for the constraints within a molecule, which helps users to determine optimal protein folds [CKH*04]. Nowadays, with cheaper hardware, better graphics cards, and faster computers, haptic steering has become very attractive [SKVS10] and can be applied even to systems comprising more than one million atoms [DPT*13].

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. Modelling of coupled molecular reactions is, thus, one of the research foci in systems biology. The most widely used tools include CellDesigner [FMKT03], VCell [MSS*08], ToxCell [CBS09], BioNetCAD [RFD*10], Rulebender [SXS*11], NetworkViewer [CAZMS14], and Cytoscape [SMO*03]. Besides visualizing the quantitative change of reactants in time-intensity curve plots, these tools offer various network visualizations. These range from following the Systems Biology Graphical Notation [LN*09] to illustrative textbook-like depictions of the modelled 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. [FKRE09] propose several methods to visually emphasize interesting aspects of particle-based cellular simulations like particle trajectories. MCell simulations can be visually inspected
Particle-based models have also been employed in visualization of polymerization where reactions add building blocks onto existing polymers [GIL*10]. Kolesár et al. [KPV*14] 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.

6.5. Visualization of quantum effects

Understanding details of reactions requires quantum chemical studies, i.e. analysis of the electronic structure of molecules by computing the ground state, the excited states, and the transition states that occur during chemical reactions. For an elementary introduction see, e.g. [Heh03]. The resulting data 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 fields and multi-fields is required. It has been demonstrated that visualization of such fields helps to reveal rich and surprising phenomena (occurring even in simplest molecular systems) [BHI*09, ABB*11, HKM*11]. Topological visual analysis of electron density fields provides information about the spatial domains attributed to individual atoms [Bad90]. There are a number of methods and tools to visually analyze covalent and non-covalent bonds [GBCG*14], weak interactions [JKMS*10, CGJK*11], and molecular orbitals [SSH*09] (see Figure 16) as well as related electron densities [HG08]. Also visualization of the resulting fuzzy molecular surfaces using volume rendering has been proposed [KCL*13].

The understanding of photoelectron transfer processes in molecular systems also requires QM approaches. For a recent example presenting tools to visualize and analyze such process, see, e.g., [GHZ*15]. In photosynthetic systems typically electronic and vibrational degrees of freedom are coupled to transfer the energy between chlorophylls; in addition QM entanglement and coherences between different parts of molecular complexes play...
a role for functioning in photosynthesis [KK12]. It is obvious that such complex spatio-temporal processes can be understood (and related to experimental results from multidimensional femtosecond spectroscopy) only with the help of advanced visualization techniques like the one presented by Sener et al. [SSB*14].

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 has ceased. On the other hand, there are several very successful and robust systems that are commonly used by domain experts both for visual analysis in their research and for dissemination of results. We decided to categorize the existing systems to four groups: freely available functionally rich systems integrating some of the state-of-the-art methods, open-source prototype tools focused on efficient algorithms and extendability, commercial systems, and web-based solutions. This section is structured with respect to this categorization.

The first category contains robust and popular tools, such as VMD [HDS96], PyMOL [DeL02], Chimera [PGH*04], YASARA View [KV14], or CAFER Analyst [KSS*14]. 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. The molecular orbitals (see Figure 16) can be computed and visualized for dynamic data using GPU-accelerated algorithms [SHLK11]. 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, isocurves, 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 [SN00], Molecet [PL00], Gabedit [All11], GaussView [DKM09], Chemcraft [And15], and Avogadro [HCL*12]. All these tools, as well as VMD and Chimera, are also able to visualize molecular orbitals that are either read from cube files (which are output by tools like the Gaussian [FTS*09] or GAMESS-US [SBB*93]) or computed directly by the visualization tools.

The second group of systems is formed by single-purpose or prototype tools, which are also freely available (most of them are open-source). The greatest advantage of such systems is that they focus on very efficient implementations with respect to latest advances in molecular visualization and rendering. One example is the QuiteMol tool by Tarini et al. [TCM06], 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 [Web09]. Other tools are released in the form of a prototype, sometimes as an open-source project that allows other developers to contribute. MegaMol by Grottell et al. [GKM*15] is an open-source rapid prototyping framework that is tailored to molecular visualization. To enable the development of novel, efficient visualization methods, it is designed as a thin supporting layer on top of the OpenGL API. Developers can 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 [KBE08], molecular surfaces [KBE09, KGE11, KSES12], and accelerated rendering and shading methods [GRDE10, GKE12]. UnityMol [LTDS*13], 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 [CVT*11] 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, which exhibit particular properties such as an increased number of triangles and required draw calls, compared to typical video game objects. Recently, UnityMol has been extended to prototype visualizations of carbohydrate molecules [PTIB14] and to act as interface for interactive molecular simulations. Furthermore, UnityMol provides a free and open starting point for video game developers and scientists who want to use molecular objects in Unity3D projects. An example of a prototype tool based on a 3D modelling and animation software is BioBlender [ACZ*12]. It is a multi-platform add-on for Blender, aiming at providing tools for the import and elaboration of biological molecules. Molecular Maya [mma] is a software toolkit that enables to import, build, and animate molecular structures in the professional Maya tool by Autodesk. One of the latest software tools designed to assemble large scale molecular models consisting of building blocks is called cellPack [JAAA*15].

The third category of systems is formed by commercial solutions like MolSoft ICM-Pro [ATK94] or Amira [SWH05]. There are also several commercial extensions for YASARA [KV14]. These
systems partially incorporate the abovementioned state-of-the-art techniques. Amira, for example, provides all the classical representations like ball-and-stick, space-filling, and cartoon representations. Furthermore, molecular surfaces like vDW surface, SAS, SES, and MSS can be rendered using GPU-based ray casting \[LBPH10\]. 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.

The last category contains web-based solutions for molecular rendering. Although such tools usually cannot integrate the latest state-of-the-art techniques covered by this report due to technical limitations, it is worthwhile to mention them because they are nowadays capable of interactive visualization of large molecular complexes. They can be embedded into web sites to provide specialized visualizations of entries in structural databases or results of structure-related calculations. One of the still most widely used web-based tools is the Java-applet Jmol \[jmo09\]. It supports loading many file formats, rendering molecular surfaces, orbitals, schematic cartoons, and other features. OpenAstexViewer \[Har02\] is another Java-based program which aims to assist in structure-based drug design. It can be used both as applet and as a standalone application. Among other functions, it offers shaded molecular surfaces with transparency and property mapping. JSmol \[HPR*13\], an extension of Jmol that uses only HTML5 and JavaScript instead of Java, is currently under development. NGL viewer \[RH15\] and iview \[LLNW14\] are examples for modern, feature-rich web-based tools utilizing WebGL, which enables hardware-accelerated rendering in the browser. The molecular visualization library 3Dmol.js \[RK15\] also uses JavaScript and WebGL. It also supports most standard representations of the molecule, including semi-transparent molecular surfaces and visualization of orbitals. As mentioned above, the established web-based visualization tools rely on triangle-based rendering due to the limitations of web-based graphics. Recently, Mwalongo et al. \[MKK*14, MKB*15\] showed that WebGL enables GPU-based ray casting in the browser. Such technological advances will probably lead to more advanced web-based molecular visualizations in the near future.

8. Conclusion and Future Challenges

Molecular biology is a very diverse field, which implies that the molecular visualization is diverse as well. Thus, a plethora of different representations—each of them having particular advantages and disadvantages—have been developed using a wide range 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 recent years has been to use GPUs not only for rendering but also for accelerating the underlying computations \[CLK*11\]. 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 advances in hardware development nowadays rather increase the degree of parallelism than the clock speed, pushing the limits of parallel computing is an important issue. This includes the efficient exploitation of 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 can alleviate the \textit{in situ} analysis of large systems.

Of equal importance are advances in the development of efficient simulation algorithms. Molecular simulations are becoming increasingly faster; new simulation techniques revealing the essential dynamics of molecular systems enable interactive simulation steering. Visually steered molecular simulations will certainly become a game changer. They will enable structural biologists to investigate the relevant aspects of complex molecular processes by interactively changing parameters as well as initial and boundary conditions.

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 \[MHLK05\], 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 \[SVS13\]. Recently, Knoll et al. \[KWN*14\] presented a parallel interactive volume ray casting of radial basis functions on CPUs.

From a more general perspective, biomolecular visualization will have to handle three major challenges: depicting physical phenomena in more detail, improving the perceptual and cognitive efficiency of visualizations, as well as depicting longer trajectories of larger molecular systems. All this will increase the significance of visual insight methods.

Regarding the first topic, instead of just depicting the molecular dynamics on a purely phenomenological level, the physical and chemical \textit{causes} for molecular events should also be visualized—both on the classical and quantum mechanical level. As compute clusters and simulation methods are improved, the number of quantum mechanical degrees of freedom that can be dealt with will increase. Therefore, novel visualization methods for the depiction of quantum phenomena in dynamic molecular systems will be needed.

Techniques improving the depth perception for complex molecular structures have been investigated extensively already (see Section 5). However, there are still opportunities to augment current visual representations with additional cues (e.g. \[SVGR15\]). Visual clutter can be addressed by developing new illustrative visualization techniques, such as specialized cutaways, unfolding, or exploded views.

Regarding increased size of input data sets, one has to deal with two problems: First, dealing with ever longer molecular trajectories; for this, new techniques will be needed, similarly to those used in interactive video analysis and video processing. Second, dealing with larger molecular systems; for this, new visual representations

© 2016 The Authors

Computer Graphics Forum published by John Wiley & Sons Ltd.
of the data will be required and, in consequence, a complete visual language for biomolecular systems needs to be established. This includes abstractions that go far beyond the level of single molecules. Today atomistic representations are available for viruses; soon small bacterial organisms will be modelled in atomic detail. When zooming out from a molecule to see the entire structure at some point, all the molecules in the model create a salt-and-pepper noise pattern without any strong informative insight. Currently there is no abstraction mechanism that would meaningfully convey these levels, as for example cartoon representations do for secondary structures. Maybe for such large molecular complexes we will be soon witnessing investigation in a meaningful definition and visual representation for quinary structure and even beyond?

Acknowledgements
This work was supported through grants from the German Research Foundation (DFG) as part of SFB 716, the Senate Department for Economics, Technology and Research in Berlin, 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 VRGI11-010, the OeAD ICM through project CZ 17/2015, the Norway grants project NF-CZO7-MOP-2-086-2014 and the PhysioIllustration research project 218023 funded by the Norwegian Research Council.

References
[AKM14] Al-Khalili, J., McFadden, J.: Life on the Edge: The Coming of Age of Quantum Biology. Random House, New York, 2014.

[All11] Allouche, A.-R.: Gabedit - A graphical user interface for computational chemistry software. Journal of Computational Chemistry 32, 1 (2011), 174–182.

[AM06] Adcock, S. A., McCammon, J. A.: Molecular dynamics: survey of methods for simulating the activity of proteins. Chemical Review 106, 5 (2006), 1589–1615.

[And15] Andrienko, G. A.: Chemcraft—graphical program for working with quantum chemistry results, 2015. Current version: 1.8, chemcraftprog.com, (Online, Sep 2016).

[AP09] Abrahamsson, E., Plotkin, S.: Biovec: A program for biomolecule visualization with ellipsoidal coarse-graining. Journal of Molecular and Graphical Modelling 28, 2 (2009), 140–145.

[ATK94] Abagyan, R., Totrov, M., Kuznetsov, D.: ICM—A new method for protein modelling and design: Applications to docking and structure prediction from the distorted native conformation. Journal of Computational Chemistry 15, 5 (1994), 488–506.

[Bad90] Bader, R.: Atoms in Molecules. Wiley & Sons, 1990.

[BD12] Brandt, S., Dahmen, H. D.: The Picture Book of Quantum Mechanics. Springer, New York, 2012.

[BDF15] Bartol, T. M., Dittrich, M., Faeder, J. R.: Mcell. In Encyclopedia of Computational Neuroscience (New York, 2015), Jaeger, D., Jung, R., (Eds.), Springer, pp. 1673–1676.

[BDST04] Baijai, C., Dieu, P., Siddavanahalli, V., Thane, A.: Texmol: interactive visual exploration of large flexible multi-component molecular complexes. In IEEE Visualization (2004), pp. 243–250.

[Ber07] Berry, D.: Molecular animation of cell death mediated by the fas pathway. Science Signaling 2007, 380 (2007).

[BGB*08] Bidmon, K., Grottel, S., Böß, F., Pleiss, J., Ertl, T.: Visual abstractions of solvent pathlines near protein cavities. Computer Graphics Forum 27, 3 (2008), 935–942.

[BHI*09] Barth, I., Hege, H.-C., Ikeda, H., Kenfack, A., Kopf, Z., Pati, A. K.: Quantum Aspects of Life. Imperial College Press, London, 2008.
G., (Eds.), vol. 321. Springer, Berlin-Heidelberg, 2010, pp. 211–227.

[DCS12] Dahl, A. C. E., Chavent, M., Sansom, M. S. P.: Bendix: intuitive helix geometry analysis and abstraction. Bioinformatics 28, 16 (2012), 2193–2194.

[DeL02] DeLano, W.: PyMOL: An open-source molecular graphics tool. CCP4 Newsletter On Protein Crystallography 40 (2002). pymol.sourceforge.net, (Online, Sep 2016).

[DSKMK13] De Heras Ciechomski, P., Klann, M., Magne, R., Koepp, H.: From biochemical reaction networks to 3D dynamics in the cell: The ZygCell3D modeling, simulation and visualization framework. In IEEE Symposium on Biological Data Visualization (2013), pp. 41–48.

[DKM09] Dennington, R., Keith, T., Millam, J.: GaussView Version 5, 2009. Semichem Inc., Shawnee Mission, KS.

[DLB08] Durrieu, M. P., Lavery, R., Baaden, M.: Interactions between neuronal fusion proteins explored by molecular dynamics. Biophysical Journal 94, 9 (2008), 3436–3446.

[DPT*13] Dreher, M., Puzzi, M., Turk, A., Chavent, M., Baaden, M., Férèy, N., Lmet, S., Raffin, B., Robert, S.: Interactive molecular dynamics: Scaling up to large systems. Procedia Computer Science 18 (2013), 20–29.

[DR13] Diecherchi, S., Rocchia, W.: A general and robust ray-casting-based algorithm for triangulating surfaces at the nanoscale. PLoS ONE 8, 4 (2013), e59744.

[DRSR15] Darboux, S. M., Rumpf, R. W., Schindhelm, A. D., Ray, W. C.: MoFlow: visualizing conformational changes in molecules as molecular flow improves understanding. In BMC Proc. (2015), vol. 9, suppl 6, p. S5.

[Ede99] Edelsbrunner, H.: Deformable smooth surface design. Discrete Computational Geometry 21, 1 (1999), 87–115.

[EEK*14] Ertl, T., Kröne, M., Kesselheim, S., Scharnowski, K., Reina, G., Holm, C.: Visual analysis for space-time aggregation of biomolecular simulations. Faraday Discussions 169 (2014), 167–178.

[EM94] Edelsbrunner, H., Mücke, E. P.: Three-dimensional alpha shapes. ACM Transactions on Graphics 13, 1 (1994), 43–72.

[ESH13] Eichelbaum, S., Scheuermann, G., Hlawitschka, M.: PointAO - improved ambient occlusion for point-based visualization. In EuroVis 2013 - Short Paper (2013), pp. 13–17.

[FKE13] Falk, M., Krone, M., Ertl, T.: Atomicistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. Computer Graphics Forum 32, 8 (2013), 195–206.

[FKRE09] Falk, M., Klann, M., Reuss, M., Ertl, T.: Visualization of signal transduction processes in the crowded environment of the cell. In IEEE Pacific Visualization Symposium (PacificVis) (2009), pp. 169–176.

[FKRE10] Falk, M., Klann, M., Reuss, M., Ertl, T.: 3D visualization of concentrations from stochastic agent-based signal transduction simulations. In IEEE International Symposium on Biomedical Imaging: From Nano to Macro (2010), pp. 1301–1304.

[FMKT03] Funahashi, A., Morohashi, M., Kitano, H., Tanimura, N.: CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. BIOSTILICO J, 5 (2003), 159–162.

[FPE*89] Fuchs, H., Pollton, J., Eyles, J., Greer, T., Goldfeather, J., Ellsworth, D., Molnar, S., Turk, G., Tebbs, B., Israel, L.: Pixel-planes 5: A heterogeneous multiprocessor graphics system using processor-enhanced memories. In SIGGRAPH 89 (1989), ACM, pp. 79–88.

[Fra02] Franoceur, E.: Cyrus Levinthral, the Kluge and the origins of interactive molecular graphics. Endeavour 26, 4 (2002), 127–131.

[FSTR13] Fioravante, M., Shook, A., Thorpe, I., Rheingans, P.: Visualizing Motional Correlations in Molecular Dynamics using Geometric Deformations. Computer Graphics Forum 32, 3pt3 (2013), 311–320.

[FTS*09] Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., Scalmani, G., Barone, V., Mennucci, B., Petersson, G. A., Nakatsuji, H., Caricato, M., Li, X., Hratchian, H. P., Izmaylov, A. F., Bloino, J., Zheng, G., Sonnenberg, J. L., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Vreven, T., Montgomery, J. A., Peralta, Jr., J. E., Ogliaro, F., Bearpark, M., Heyd, J. J., Brothers, E., Kudin, K. N., Staroverov, V. N., Kobayashi, R., Normand, J., Raghavachari, K., Rendell, A., Burant, J. C., Iyengar, S. S., Tomasi, J., Cossi, M., Rega, N., Millam, J. M., Klene, M., Knox, J. E., Cross, J. B., Bakken, V., Adamo, C., Jaramillo, J., Pople, J., Ortmann, R., Stratmann, R. E., Yazyev, O., Austin, A. J., Cammi, R., Pomelli, C., Ochterski, J. W., Martin, R. L., Morokuma, K., Zakrzewski, V. G., Voth, G. A., Salvador, P., Dannenberg, J. J., Dapprich, S., Daniels, A. D., Farkas, O., Foreman, J. B., Ortiz, J. V., Cioslowski, J., Fox, D. J.: Gaussian 09 Revision D.01, 2009. Gaussian Inc., Wallingford, CT.

[GB78] Greer, J., Bush, B. L.: Macromolecular shape and surface maps by solvent exclusion. Proceedings of the National Academy of Science of the United States of America 75 (1978), 303–307.

[GBGC*14] Günter, D., Boto, R., Contreras-Garcia, J., Pique-Mal, J.-P., Tierny, J.: Characterizing molecular interactions in chemical systems. IEEE Transactions on Visualization and Computer Graphics 20, 12 (2014), 2476–2485.

[GF07] Goodard, T. D., Ferrin, T. E.: Visualization software for molecular assemblies. Current Opinion in Structural Biology 17, 5 (2007), 587–95.
B. Kozlíková et al. / Visualization of Biomolecular Structures

(KV14) Krieger, E., Vriend, G.: YASARA view–molecular graphics for all devices—from smartphones to workstations. Bioinformatics 30, 20 (2014), 2981–2982.

(KV15) Krieger, E., Vriend, G.: New ways to boost molecular dynamics simulations. Journal of Computational Chemistry 36, 13 (2015), 996–1007.

(KWN*14) Knoll, A., Wald, I., Navrátil, P. A., Bowden, A., Reda, K., Papka, M. E., Gaither, K.: RBF volume ray casting on multicolour and manycore CPUs. Computer Graphics Forum 33, 3 (2014), 71–80.

(LB02) Laug, P., Borouchaki, H.: Molecular surface modeling and meshing. Engineering with Computers 18, 3 (2002), 199–210.

(LBBH12) Lindsay, N., Baum, D., Bondar, A.-N., Hege, H.-C.: Dynamic channels in biomolecular systems: Path analysis and visualization. In IEEE Symposium on Biological Data Visualization (2012), pp. 99–106.

(LBH12) Lindsay, N., Baum, D., Hege, H.-C.: Interactive rendering of materials and biological structures on atomic and nanoscopic scale. Computer Graphics Forum 31 (2012), 1325–1334.

(LBH14) Lindsay, N., Baum, D., Hege, H.-C.: Ligand excluded surface: A new type of molecular surface. IEEE Transactions on Visualization and Computer Graphics 20, 12 (2014), 2486–2495.

(LPSP10) Lindsay, N., Baum, D., Prohaska, S., Hege, H.-C.: Accelerated visualization of dynamic molecular surfaces. Computer Graphics Forum 29, 3 (2010), 943–952.

(LC87) Lorensen, W. E., Cline, H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In ACM SIGGRAPH '87 (1987), vol. 27, pp. 163–169.

(LCD06) Luft, T., Colditz, C., Deusen, O.: Image enhancement by unsharp masking the depth buffer. ACM Transactions on Graphics 25, 3 (2006), 1206–1213.

(LIMJ15) Luehr, N., Jin, A. G. B., Martinez, T. J.: Ab initio interactive molecular dynamics on graphical processing units (gpus). Journal of Chemical Theory and Computation 11, 10 (2015), 4536–4544.

(LKEP14) Lawonn, K., Krone, M., Ertl, T., Preim, B.: Line integral convolution for real-time illustration of molecular surface shape and salient regions. Computer Graphics Forum 33, 3 (2014), 181–190.

(LLNW14) Li, H., Leung, K.-S., Nakane, T., Wong, M.-H.: iview: an interactive WebGL visualizer for protein-ligand complex. BMC Bioinformatics 15, 1 (2014), 56–62.

(LMPAV15) Le Muzic, M., Autin, L., Parulek, J., Viola, I.: celIView: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In EG Workshop on Visual Computing for Biology and Medicine (2015), pp.

(LMPSV14) Le Muzic, M., Parulek, J., Stavrak, A., Viola, I.: Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. Computer Graphics Forum 33, 3 (2014), 141–150.

(LMWVP15) Le Muzic, M., Waldner, M., Parulek, J., Viola, I.: Illustrative timelapse: A technique for illustrative visualization of particle simulations on the mesoscale level. In IEEE Pacific Visualization Symposium (2015), pp. 247–254.

(LN*09) Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjev, M., Wimalaratne, S., Bergman, F. T., Gauges, R., Ghazal, P., Kawai, H., Li, L., Matsuoka, Y., Villégé, A., Boyd, S. E., Calzone, L., Courtot, M., Drogozou, U., Freeman, T. C., Funahashi, A., Ghosh, S., Joukaku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G., Goryanin, I., Kell, D. B., Sander, C., Sauro, H., Snoep, J. L., Kohn, K., Kitano, H.: The systems biology graphical notation. Nat. Biotechnology 27, 8 (2009), 735–741.

(LPK06) Lee, J., Park, S., Kim, J.-I.: View-dependent rendering of large-scale molecular models using level of detail. In Int. Conf. on Hybrid Information Technology (ICKIT) (2006), vol. 1, pp. 691–698.

(LR71) Lee, B., Richards, F. M.: The interpretation of protein structures: Estimation of static accessibility. Journal of Molecular Biology 55, 3 (1971), 379–380.

(LTDS*13) Lv, Z., Tek, A., Da Silva, F., Empereur-Mot, C., Chavent, M., Baaden, M.: Game on, science - how video game technology may help biologists tackle visualization challenges. PLoS ONE 8, 3 (2013), e57990.

(LV02) Lee, C. H., Varshteyn, A.: Representing thermal vibrations and uncertainty in molecular surfaces. In SPIE Conference on Visualization and Data Analysis (2002), pp. 80–90.

(LVRH07) Lampe, O. D., Viola, I., Reuter, N., Hauser, H.: Two-level approach to efficient visualization of protein dynamics. IEEE Transactions on Visualization and Computer Graphics 13, 6 (2007), 1616–1623.

(MEP92) Molnar, S., Eyles, J., Poultion, J.: Pixellow: High-speed rendering using image composition. In SIGGRAPH '92 (1992), ACM, pp. 231–240.

(MHLK05) Moll, A., Hildebrandt, A., Lenthof, H.-P., Kohlbacher, O.: BALLView: A tool for research and education in molecular modeling. Bioinformatics (2005).

(Mil94) Miller, G.: Efficient algorithms for local and global accessibility shading. In SIGGRAPH '94 (1994), pp. 319–326.

(MKB*15) Mwalongo, F., Krone, M., Becher, M., Reina, G., Ertl, T.: Remote Visualization of Dynamic Molecular Data Using WebGL. In International Conference on 3D Web Technology (2015), pp. 115–122.
[MKK*14] MWALONGO, F., KRONE, M., KARCH, G., BECHER, M., REINA, G., ERTI, T.: Visualization of Molecular Structures using State-of-the-Art Techniques in WebGL. In International Conference on 3D Web Technology (2014), vol. 19, ACM, pp. 133–141.

[MM04] MORITZ, E., MEYER, J.: Interactive 3D protein structure visualization using virtual reality. In IEEE Symposium on Bioinformatics and Bioengineering (2004), pp. 503–507.

[mm] Molecular Maya Toolkit – mMaya. www.molecularmovies.com/toolkit/, (Online, Sep 2016).

[MSS*08] MORARU, I., SCHAFI, J., SLEPCHENKO, B., BLINOV, M., MORGAN, F., LAKSHMINARAYANA, A., GAO, F., LI, Y., LOEW, L.: Virtual cell modelling and simulation software environment. IET Systems Biology 2, 5 (2008), 352–362.

[NCS85] NAMBA, K., CASPAR, D. L., STUBBS, G. J.: Computer graphics representation of levels of organization in tobacco mosaic virus structure. Science 227, 4688 (1985), 773–776.

[NHK*95] NELSON, M., HUMPHREY, W., KURFIEN, R., GURSOY, A., DALKE, A., KALE, L., SKEEL, R., SCHULTEN, K.: MDScope—a visual computing environment for structural biology. Computer Physics Communications 91, 1–3 (1995), 111–133.

[PB13] PARULEK, J., BRAMBILLA, A.: Fast blending scheme for molecular surface representation. IEEE Transactions on Visualization and Computer Graphics 19, 12 (2013), 2653–2662.

[PC51] PAULING, L., COREY, R. B.: The pleated sheet, a new layer configuration of polypeptide chains. Proceedings of the National Academy of Science of the United States of America 37, 5 (1951), 251–256.

[PCB51] PAULING, L., COREY, R. B., BRANSON, H. R.: The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. Proceedings of the National Academy of Science of the United States of America 37, 4 (1951), 205–211.

[PGH*04] PETTERSSEN, E. F., GODDARD, T. D., HUANG, C. C., COUCH, G. S., GREENBLATT, D. M., MENG, E. C., FERRIN, T. E.: UCSF Chimera - a visualization system for exploratory research and analysis. Journal of Computational Chemistry 25, 13 (2004), 1605–1612.

[Pho75] PHONG, B. T.: Illumination for computer generated pictures. Communications of the ACM 18, 6 (1975), 311–317.

[PJR*14] PARULEK, J., JÖNSSON, D., ROPINSKI, T., BRUCKNER, S., YNEBERMAN, A., VIOLA, I.: Continuous levels-of-detail and visual abstraction for seamless molecular visualization. Computer Graphics Forum 33, 6 (2014), 276–287.

[PL00] PORTMANN, S., LUTHI, H. P.: MOLEKEL: An interactive molecular graphics tool. Chima 54, 12 (2000), 766–770.

[PS05] PUMPON, S. J., SLEPOY, A.: Microbial cell modeling via reacting diffusive particles. Journal of Physics: Conference Series 16 (2005), 305–309.

[PS09] POSTARNAKEVICH, N., SINGH, R.: Global-to-local representation and visualization of molecular surfaces using deformable models. In ACM Symposium on Applied Computing (2009), pp. 782–787.

[PS15] PATANE, G., SPAGNUOLO, M.: State-of-the-art and perspectives of geometric and implicit modeling for molecular surfaces. In Computational Electrostatics for Biological Applications, Rochester, W., Spagnuolo, M., (Eds.). Springer, 2015, pp. 157–176.

[PTIB14] PÉREZ, S., TUBIANA, T., IBERTY, A., BAADEN, M.: Three-dimensional representations of complex carbohydrates and polysaccharides—SweetUnityMol: A video game-based computer graphic software. Glycobiology (2014).

[PV12] PARULEK, J., VIOLA, I.: Implicit representation of molecular surfaces. In IEEE Pacific Visualization Symposium (PacificVis) (2012), pp. 217–224.

[RRBT14] ROZMANOV, D., BAOUKINA, S., TIELEMAN, D. P.: Density-based visualization for molecular simulation. Faraday Discussions 169, 0 (2014), 225–243.

[RCDFO8] RUSINKIEWICZ, S., COLE, F., DI CARLO, D., FINKELSTEIN, A.: Line drawings from 3D models. In ACM SIGGRAPH 2008 Classes (2008), vol. 39, pp. 1–356.

[RCK09] RYU, J., CHO, Y., KIM, D.-S.: Triangulation of molecular surfaces. Computer Aided Design 41, 6 (2009), 463–478.

[RE05] REINA, G., ERTI, T.: Hardware-accelerated glyphs for mono- and dipoles in molecular dynamics visualization. In EG/IEEE VGTG Symposium on Visualization (2005), pp. 177–182.

[RFD*10] RIALE, S., FELICORI, L., DIAS-LOPES, C., PÉRES, S., EL ATIA, S., THIERRY, A. R., AMAR, P., MOLINA, F.: BioNetCAD: design, simulation and experimental validation of synthetic biochemical networks. Bioinformatics 26, 18 (2010), 2298–2304.

[RH15] ROSE, A., HILDERBRAND, P.: NGL Viewer: a web application for molecular visualization. Nucleic Acids Research 43, 5 (2015), W576–9.

[Ric77] RICHARDS, F. M.: Areas, volumes, packing, and protein structure. Annual Review of Biophysics and Bioengineering 6, 1 (1977), 151–176.

[Ric81] RICHARDSON, J.: The anatomy and taxonomy of protein structure. Advances in Protein Chemistry 34 (1981), 167–339.

[RJ99] RHEINGANS, P., JOSHI, S.: Visualization of molecules with positional uncertainty. In Data Visualization ’99 (1999), pp. 299–306.

[RK15] RIGO, N., KOES, D.: 3Dmol.js: molecular visualization with WebGL. Bioinformatics 31, 8 (2015), 1322–1324.

[RRS63] RAMACHANDRAN, G., RAMAKRISHNAN, C., SASISEKHARAN, V.: Stereochemistry of polypeptide chain configurations. Journal of Molecular Biology 7, 1 (1963), 95–99.
References

[RS07] Rahi, S. J., Sharp, K.: Mapping complicated surfaces onto a sphere. International Journal of Computational Geometry and Applications 17, 4 (2007), 305–329.

[RSLS13] Roberts, E., Stone, J. E., Luthey-Schulten, Z.: Lattice Microbes: high-performance stochastic simulation method for the reaction-diffusion master equation. Journal of Computational Chemistry 34, 3 (2013), 245–255.

[SB01] Stiles, J. R., Barton, T. M.: Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology Using MCell. CRC Press, 2001, ch. 4.

[SBB+93] Schmidt, M. W., Baldrige, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koceki, S., Matsunaga, N., Nguyen, K. A., Su, S., Schmidt, M. W., Baldrige, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koceki, S., Matsunaga, N., Nguyen, K. A., Su, S. J., Windsus, T. L., Dupuis, M., Montgomery, J. A.: General atomic and molecular electronic structure system. Journal of Computational Chemistry 14, 11 (1993), 1347–1363.

[Sch26] Schrödinger, E.: Quantisierung als Eigenwertproblem - vierte Mitteilung. Annalen der Physik 81, 18 (1926), 109–139.

[SEBH02] Schmidt-Ehrenberg, J., Baum, D., Hege, H.-C.: Visualizing dynamic molecular conformations. In IEEE Visualization (2002), pp. 235–242.

[SF02] Smith, B., Frenkel, D.: Understanding molecular simulation: from algorithms to applications. Elsevier Science & Technology, USA, 2002.

[SGB+14] Shaw, D. E., Grossman, J. P., Bank, J. A., Batson, B., Adam Butts, J., Chao, J. C., Deneroff, M. M., Dior, R. O., Even, A., Fenton, C. H., Forte, A., Gaagliardo, J., Gill, G., Greskamp, B., Hoe, C. R., Ierardi, D. J., Iserovich, L., Kuskin, J. S., Larson, R. H., Li-Siang Lee, T. L., Lerner, A. K., Li, C., Killebrew, D., Mackenzie, K. M., Mok, S. Y. H., Moraes, M. A., Mueller, R., Nocilolo, L. J., Peticolas, J. L., Quan, T., Ramot, D., Salmon, J. K., Scarpazza, D. P., Schafer, U. B., Siddique, N., Snyder, C. W., Ping Tak Peter Tang, J. S., Theobald, M., Tomia, H., Towles, B., Vitale, B., Wang, S. C., Young, C.: 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 (2014), SC’14, pp. 41–53.

[SGG15] Staub, J., Grottell, S., Gunhold, S.: Visualization of particle-based data with transparency and ambient occlusion. Computer Graphics Forum 34, 3 (2015), 151–160.

[SGG01] Stone, J. E., Gullingsrud, J., Schulten, K., Grayson, P.: A system for interactive molecular dynamics simulation. In ACM Symposium on Interactive 3D Graphics (2001), pp. 191–194.

[She99] Sherstyuk, A.: Kernel functions in convolution surfaces: a comparative analysis. The Visual Computer 15, 4 (1999), 171–182.

[SHL11] Stone, J. E., Hardy, D. J., L. V. K., K. S.: GPU-accelerated computation and interactive display of molecular orbitals. In GPU Computing Gems Emerald Edition, Hwu, W.-m. W., (Ed.). Morgan Kaufmann, 2011, ch. 1, pp. 5–18.

[SI12] Szecsi, L., Illies, D.: Real-time metaball ray casting with fragment lists. In EG 2012 Short Papers (2012), pp. 93–96.

[SKS+13] Scharnowski, K., Krone, M., Sadlo, F., Beck, P., Roth, J., Trebin, H.-R., Ertl, T.: 2012 IEEE Visualization Contest Winner: Visualizing polarization domains in barium titanate. IEEE Computer Graphics and Applications 33, 5 (2013), 9–17.

[SKVS10] Stone, J. E., Kohlmeier, A., Vondivort, K. L., Schulten, K.: Immersive molecular visualization and interactive modeling with commodity hardware. In Advances in Visual Computing, vol. 6454 of Lecture Notes in Computer Science. Springer, 2010, pp. 382–393.

[Smi60] Smith, D. K.: Bibliography on molecular and crystal structure models. Tech. rep., US National Bureau of Standards, 1960.

[SMM+03] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 13, 11 (2003), 2498–2504.

[SMW95] Sayle, R. A., Milner-White, E.: RASMOL: biomolecular graphics for all. Trends in Biochemical Science 20, 9 (1995), 374–376.

[SN00] Schaftenaar, G., Noordik, J.: Molden: a pre- and post-processing program for molecular and electronic structures. Journal of Computer Aided Molecular Design 14, 2 (2000), 123–134.

[SOS97] Sanner, M. F., Olson, A. J.: Real time surface reconstruction for moving molecular fragments. In Pacific Symposium on Biocomputing (1997), pp. 385–396.

[SOS96] Sanner, M. F., Olson, A. J., Speghner, J.-C.: Reduced surface: An efficient way to compute molecular surfaces. Biopolymers 38, 3 (1996), 305–320.

[SRRB94] Surles, M. C., Richardson, J. S., Richardson, D. C., Brooks, F. P.: Sculpting proteins interactively: Continual energy minimization embedded in a graphical modeling system. Protein Science 3, 2 (1994), 198–210.

[SSB+14] Sener, M., Stone, J. E., Barragan, A., Singharoy, A., Teo, I., Vondivort, K. L., Isakowitz, B., Liu, B., Goh, B. C., Phillips, J. C., Kourkoutis, L. F., Hunter, C. N., Schulten, K.: Visualization of energy conversion processes in a light harvesting organelle at atomic detail. SC ‘14 Visualization and Data Analytics Showcase, 2014.

[SSH+09] Stone, J. E., Saam, J., Hardy, D. J., Vondivort, K. L., Hwu, W.-m. W., Schulten, K.: High performance computation

© 2016 The Authors

Computer Graphics Forum published by John Wiley & Sons Ltd.
and interactive display of molecular orbitals on GPUs and multicore CPUs. In Workshop on General Purpose Processing on Graphics Processing Units (2009), pp. 9–18.

[SLS16] Stone, J. E., Sherman, W. R., Schulten, K.: Immersive Molecular Visualization with Omnidirectional Stereoscopic Ray Tracing and Remote Rendering. In IEEE International Parallel and Distributed Processing Symposium Workshop (2016).

[ST90] Saito, T., Takahashi, T.: Comprehensible rendering of 3-D shapes. Computer Graphics (Proc. SIGGRAPH 1990) 24, 4 (1990), 197–206.

[Str95] Stryer, L.: Biochemistry. W.H. Freeman and Company, New York, 1995.

[SVGR15] Skanberg, R., Vazquez, P.-P., Guallar, V., Ropinski, T.: Real-time molecular visualization supporting diffuse illumination and ambient occlusion. IEEE Transactions on Visualization and Computer Graphics 22 (2015).

[SVS13] Stone, J. E., Vandivort, K. L., Schulten, K.: GPU-accelerated molecular visualization on petascale supercomputing platforms. In Int. Workshop on Ultrascale Visualization (UltraVis ’13) (2013), pp. 6:1–6:8.

[SWBG06] Sigg, C., Weyrich, T., Botsch, M., Gross, M.: GPU-based ray-casting of quadratic surfaces. In EG Symposium on Point-Based Graphics (2006), pp. 59–65.

[SWH05] Stallings, D., Westerhoff, M., Hege, H.-C.: Amira: a highly interactive system for visual data analysis. In The Visualization Handbook. Elsevier, 2005, pp. 749–767.

[SXS*11] Smith, A., Xu, W., Sun, Y., Faeder, J., Marai, G.: RuleBender: Integrated visualization for biochemical rule-based modeling. In IEEE Symposium on Biological Data Visualization (2011), pp. 1–8.

[TA96] Tatchev, M., Abagyan, R.: The contour-buildup algorithm to calculate the analytical molecular surface. Journal of Structural Biology 116, 1 (1996), 138–143.

[TCB+12] Tek, A., Chavent, M., Baaden, M., Delalande, O., Bourdot, P., Ferey, N.: Advances in human-protein interaction—interactive and immersive molecular simulations. In Protein-Protein Interactions—Computational and Experimental Tools (2012), Cai, W., Hong, H., (Eds.), InTech. Chapter 2.

[TMC06] Tarini, M., Cignoni, P., Montani, C.: Ambient occlusion and edge cueing for enhancing real time molecular visualization. IEEE Transactions on Visualization and Computer Graphics 12, 5 (2006), 1237–1244.

[Tha05] Thaller, B.: Advanced Visual Quantum Mechanics. Springer, New York, 2005.

[Toz05] Tozzini, V.: Coarse-grained models for proteins. Current Opinion in Structural Biology 15, 2 (2005), 144–150.

[TWK*11] Tomàs B., Walter, J., Krone, M., Hasse, H., Ertl, T.: Interactive exploration of polymer-solvent interactions. In Vision, Modeling and Visualization (2011), vol. 16, pp. 301–308.

[UD10] Uversky, V. N., Dunker, A. K.: Understanding protein non-folding. Biochimica et Biophysica Acta - Proteins and Proteomics 1804, 6 (2010), 1231–1264.

[VBW94] Varshney, A., Brooks, F. P., Wright, W. V.: Linearly scalable computation of smooth molecular surfaces. IEEE Computer Graphics and Applications 14, 5 (1994), 19–25.

[vdW73] Van der Waals, J. D.: Over de continuiteit van den gas-en vloeistofstoestand. PhD thesis, Univ. Leiden, 1873.

[vdZLB11] van der Zwan, M., Lueks, W., Bekker, H., Isenberg, T.: Illustrative molecular visualization with continuous abstraction. Computer Graphics Forum 30, 3 (2011), 683–690.

[VFG98] Velho, L., Figueiredo, L. H. d., Gomes, J. A.: Implicit Objects in Computer Graphics. Springer, New York, 1998.

[VFSL02] Vázquez, P.-P., Feixas, M., Sbert, M., Lloret, A.: Viewpoint entropy: a new tool for obtaining good views of molecules. In Symposium on Data Visualisation (VISSYM ’02) (2002), vol. 22, pp. 183–188.

[VHG*M00] van Heel, M., Gowen, B., Matadeen, R., Orlova, E. V., Finn, R., Pape, T., Cohen, D., Stark, H., Schmidt, R., Schatz, M., Patwardhan A.: Single-particle electron cryo-microscopy: towards atomic resolution. Quarterly Reviews of Biophysics 33, 4 (2000), 307–369.

[VPL11] Vehlow, C., Preim, B., Lappe, M.: Visualization of anisotropic contact potentials within protein structures. In IEEE Symposium on Biological Data Visualization (2011), pp. 31–38.

[WB11] Wahle, M., Birmanns, S.: GPU-accelerated visualization of protein dynamics in ribbon mode. In SPIE Conference on Visualization and Data Analysis (2011), pp. 786–805.

[WC53] Watson, J., Crick, F.: A structure for deoxyribose nucleic acid. Nature 421, 6921 (1953), 397–3988.

[Web09] Weber, J. R.: ProteinShader: Illustrative rendering of macromolecules. BMC Structural Biology 9, 19 (2009).

[Woo97] Woolfson, M. M.: An Introduction to X-ray Crystallography. Cambridge University Press, Cambridge, 1997.

[WQC*15] Wang, L., Qiao, H., Cao, C., Xu, S., Zou, S.: An accurate model for biomolecular helices and its application to helix visualization. PLoS ONE 10, 6 (2015), 1–13.

[Wüt86] Wüthrich, K.: NMR of proteins and nucleic acids. John Wiley & Sons, New York, 1986.

[WW15] Wahle, M., Wreggers, W.: Multi-scale visualization of molecular architecture using real-time ambient occlusion

© 2016 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.
in Sculptor. *PLoS Computational Biology* 11, 10 (2015), e1004516.

[Yu09] Yu, Z.: A list-based method for fast generation of molecular surfaces. In *Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (2009), vol. 31, pp. 5909–5912.

[ZIK98] Zhukov, S., Jones, A., Kronin, G.: An ambient light illumination model. In *Eurographics Workshop on Rendering* (1998), pp. 45–56.

[ZXB07] Zhao, W., Xu, G., Bajaj, C.: An algebraic spline model of molecular surfaces. In *Symposium on Solid and Physical Modeling (SPM)* (2007), pp. 297–302.

**Supporting Information**

Additional Supporting Information may be found in the online version of this article at the publisher’s web site:

**Timeline**