Seamless Visual Abstraction of Molecular Surfaces

Julius Parulek [*]
University of Bergen, Norway

Timo Ropinski[†]
Linköping University, Sweden

Ivan Viola[‡]
Vienna University of Technology, Austria
University of Bergen, Norway

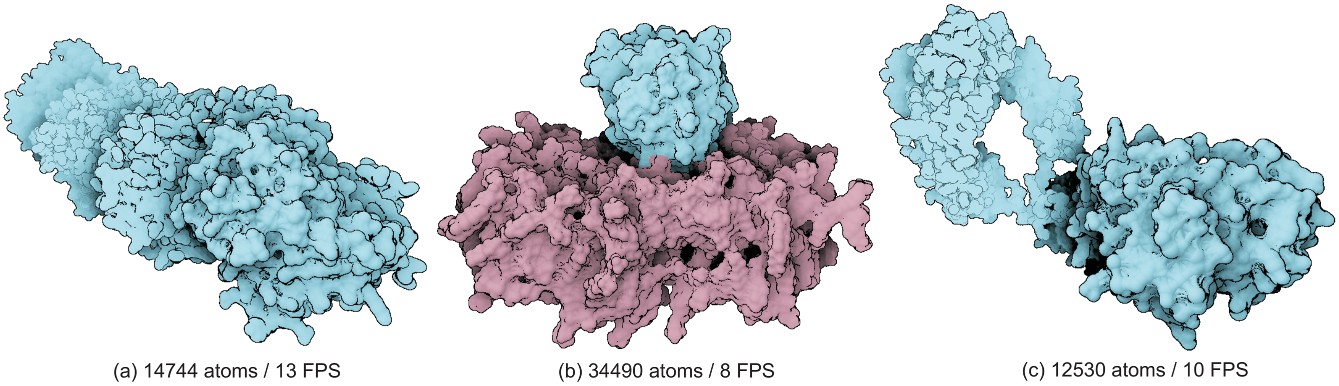

Figure 1: Three molecular examples demonstrating utilization of our seamless visual abstraction. We employ three different geometric representations (solvent-excluded surface, Gaussian kernels and van der Waals spheres) and their corresponding shading abstractions (diffuse shading and contours, constant shading with contours, constant shading without contours). The application of individual levels is based on the distance to the camera; i.e., the closest surface is based on highest geometrical and shading levels while the farthest are displayed via the lowest ones. In the presented examples we achieved $5\times - 10\times$ speed-up as compared to the SES representation only. a) Tubulin: RB3 stathmin-like domain complex. b) The phospholipase bound the lipid membrane. c) Immunoglobulin.

Abstract

Molecular visualization is often challenged with rendering of large sequences of molecular simulations in real time. We introduce a novel approach that enables us to show even large protein complexes over time in real-time. Our method is based on the level-of-detail concept, where we exploit three different molecular surface models, solvent excluded surface (SES), Gaussian kernels and van der Waals spheres combined in one visualization. We introduce three shading levels that correspond to their geometric counterparts and a method for creating seamless transition between these representations. The SES representation with full shading and added contours stands in focus while on the other side a sphere representation with constant shading and without contours provide the context. Moreover, we introduce a methodology to render the entire molecule directly using the A-buffer technique, which further improves the performance. The rendering performance is evaluated on series of molecules of varying atom counts.

CR Categories: J.3 [Computer Applications]: Life and Medical Sciences—Biology and Genetics I.3.3 [COMPUTER GRAPHICS]: Picture/Image Generation—Viewing algorithms

Keywords: Visualization of Molecular Surfaces, Implicit Surfaces, Level-of-detail

1 Introduction

Molecular visualization today is challenged by molecular dynamics (MD) simulations with the requirement of displaying huge amounts of atoms at interactive frame rates for the visual analysis of binding sites. Simulated datasets do not longer consist of only one moderately sized macromolecule, but instead of molecular systems representing complex interactions, e. g., a phospholipid vesicle membrane together with proteins anchored in the membrane. One can easily obtain datasets where tens- or hundreds of thousands of atoms are animated throughout a series of 1000 time-steps. The fact that molecular dynamics can consist of even several thousands of frames, preprocessing of frames should be eliminated and the sequence should be rendered in real-time without computing any proxy structure such as an octree or other space partitioning schemes.

To analyze a binding site, a special visual representation is most popular among molecular biologists known as the solvent-excluded surface (SES) [Richards 1977]. This representation directly conveys information whether a solvent of a certain size is able to reach a particular binding site on the surface of the macromolecule. While this representation is valued by the molecular biology domain, it is also expensive to compute. To achieve interactivity with the scene, biologists sacrifice a bit of information provided by SES and investigate molecules with blobby Gauss kernel representations [Blinn 1982], or with a simple space filling approach. The latter one, for

[*] e-mail: Julius.Parulek@uib.no
[†] e-mail: timo.ropinski@liu.se
[‡] e-mail: viola@cg.tuwien.ac.at

example, can be represented very quickly by impostor-based sphere splatting, but unfortunately it does not answer precisely whether a solvent can bind at a specific location to a macromolecule. An open question remains how to provide a binding-site relevant visualization to the molecular biology domain so that analysts can obtain interactive frame-rates of molecular dynamics simulations on their desktop workstations without any visualization based precomputation. In the search for the appropriate solution we turn to the visual crafts for inspiration, which have been already successfully applied on molecular visualization [van der Zwan et al. 2011].

Illustrators sometimes take a different approach for visually abstracting molecules, or other instances of the same object, from details. Instead of modifying the molecular representation into an entirely different molecular abstraction, they effectively use the perceptual principles of object constancy to depict structures that are too far away to recognize the details, by simplified representation of that object. By such an approach the details become prominent in the structural part that is closest to the viewer, while farther parts give visual prominence to overall structure rather than individual details. Illustrators rely on the object constancy for convenience so that they do not need to depict every smallest detail of every single object instance. Thus their rendering gets faster. The smooth abstraction is also convenient for the viewer, whose cognitive processing related to object constancy autocompletes the simplified visual representation with an object instance. A beautiful utilization of this approach can be seen on Winsor McCay's artwork of "When Black Death Rode" shown in Fig. 2, which was exemplified by a professional scientific illustrator Bill Andrews [Andrews 2006].

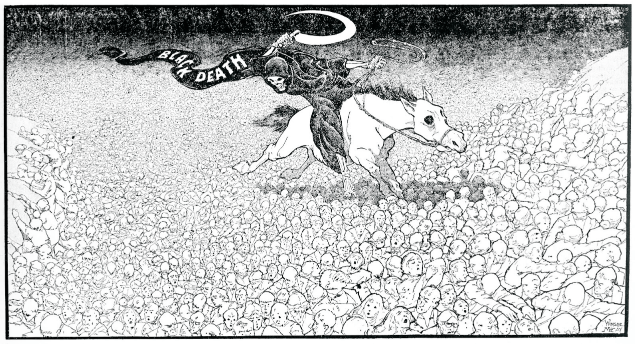

Figure 2: Object constancy employed in visual arts by Winsor McCay "When Black Death Rode".

To address the molecular visualization challenge, delineated above, we suggest that one opportunity is to employ a seamless level-of-detail on-the-fly rendering scheme in the same way as illustrators approach rendering scenes containing multiple instances of the same object and taking advantage of the object constancy perceptual principle. As a general rule, closest to the viewer we aim at providing a maximum of relevant information related to the structure and binding sites. Such information is conveyed by the SES representation. Farther away from the viewer, we are smoothly changing the visual representation to an approximation of SES through Gaussian kernels. The least detailed representation is based on simple sphere splatting and is dedicated to structures farthest away from the viewer. The thoughts about more general solution can lead us to the definition of a 3*D* importance function that can be based on the distance measure from a molecular feature. The main reason behind adopting three levels and not just two, e.g., SES and spheres, is that usually a viewer has three zones that are cognitively processed; focus, focus-relevant, and context zone. Additionally, the Gauss model provides smooth transition between SES and spheres representations (Fig. 1).

Nevertheless, the question that remains unanswered, is how we can preserve smoothness in detail-level transitions. Smoothness in transitions is an important requirement as an abrupt change in level-of-detail will become a salient artifact that will involuntarily attract the attention of the biologist. To tackle this problem, we propose to utilize the implicit surface representation, where we can seamlessly blend from one surface representation to another one, as this is an inherent property of implicit models. The seamless illustration-inspired level-of-detail scheme for molecular systems based on implicit surfaces is the main contribution of this paper. Additionally, the scheme fulfils the focus and context model, where both levels are blended via the seamless transformations. While illustrative representations have been investigated in the context of molecular visualization earlier, they have never been investigated within the context of a level-of-detail scheme.

The contributions of this paper are the following: We propose a novel visualization approach that speed up the overall rendering performance by utilization of a level-of-detail concept applied via three molecular surface models. Additionally, we introduce three different shading abstractions that are aligned with the surface representations.

2 Related Work

As this paper deals with two aspects of molecular visualization, appropriate visual representations and interactive rendering techniques, we have divided the related work into two sections accordingly.

Visual representations: Tarini et al. present a real-time algorithm for visualizing molecules with the goal to improve depth perception [Tarini et al. 2006]. By combining ambient occlusion and edge-cueing together with GPU data structures, they achieve interactive frame rates for molecules of up to the order of 10^6 atoms. Based on this representation, the authors report an improved understanding of the molecule structure. While we exploit different representations mainly in order to allow for efficient rendering, Lueks et al. combine different representations of a molecule in a single view in order to support understanding of different abstraction levels [**?**]. By allowing the user to control the seamless transition between different molecule representations, these can be viewed in a combined manner and thus reveal information at different degrees of structural abstraction. The abstractions which are combined, are based on previous work presented by van der Zwan et al. [van der Zwan et al. 2011]. The authors classify molecular representations based on their *illustrativeness*, structural abstraction and spatial perception. By giving the user control over these three parameters, s/he can change the depiction of a molecule. Thus the possible representations largely resemble known molecular representations widely used in text books. The *illustrativeness* presented by van der Zwan et al. is achieved by combining different rendering styles. Similar to the work done by Tarini et al. [Tarini et al. 2006], they also experiment with ambient occlusion techniques. In contrast, Weber presents a cartoon style rendering algorithm for protein molecules, which exploits GPU shaders to generate interactive pen-and-ink effects [Weber 2009]. Many of the presented illustration models go back to the original work done by David Goodsell [Goodsell 2009], who has developed a simplistic but expressive style for representing molecules through space filling. His approach combines ambient occlusion with cel-shading and silhouettes in order to illustrate residuals. This illustration approach has for instance been recently adopted by Falk et al. [Falk et al. 2012], and it also inspired the creation of the renderings shown in this paper.

Interactive rendering: Besides the recent efforts dealing with the visual representation of molecules, a lot of work has been dedicated to increase the overall rendering performance. With this respect, Sharma et al. present an octree-based approach, which allows to render billions of atoms interactively by exploiting view-frustum culling [Sharma et al. 2004]. During rendering a combination of probabilistic and depth-based occlusion algorithms is used to determine the visible atoms. More recently, Grottel et al. have investigated different data simplification strategies, whereby they also consider culling [Grottel et al. 2010]. In particular, they take into account data quantization, video memory based caching, and a two-level occlusion culling strategy. Lampe et al. focus on the visualization of slow dynamics for large protein assemblies [Daae Lampe et al. 2007]. To represent these large-scale dynamic models, they also use a hierarchical approach, whereby the topmost layer is representing residues being the high-level building blocks of a molecule. For each residue only orientation information is sent to the GPU, where the generation of the individual atoms is then performed on-the-fly. Since SES represents the most advanced representation of molecular surfaces, which allows to study molecule interactions and evolution, some effort has been also dedicated to improve the rendering of these fairly complex structures. Parulek and Viola propose a SES representation which is based on implicit surfaces [Parulek and Viola 2012]. By exploiting CSG operations on these surfaces, they obtain implicit functions which locally describe a molecule's surface. As their ray-casting based rendering of this representation requires no preprocessing, they are able to vary SES parameters interactively. More recently, Lindow et al. have presented a rendering technique which allows to render billions of atoms potentially represented in different scales [Lindow et al. 2012]. Similar to the work done by Lampe et al. [Daae Lampe et al. 2007], they exploit the fact that molecules are build up of repetitive structures. Together with the assumption that only a subset of the opaque molecules is visible due to occlusion, they are able to load all atomic data on the GPU, even for large models. They also conclude that level-of-detail (LOD) rendering would be the next demanding step for further improving rendering performance. Frey et al. focus on molecular dynamics simulation data [Frey et al. 2011]. In order to speed up rendering of this data, they reduce the amount of particles by focusing on those considered as relevant for the visualization. In contrast to our technique this resembles a data reduction approach instead of a data simplification approach.

3 Methodology

Motivated by the need for visualization of large molecular systems, we propose a seamless visual abstraction scheme from a most computationally expensive, but most relevant visualization technique, up to the fastest space-filling representation that is suitable for representing the context. The key technology that allows for the seamless transition, is the implicit surface representation on which all the visual abstractions are based on. We define three different levels of visual abstraction, with overlapping transition zones: a near-field, a mid-field and a far-field. The field boundaries are defined by an importance function, $t(\mathbf{p})$. Besides the distance from the viewer used as our primary example, the importance function can be thought of as a distance measure from an intersecting molecular feature (e.g., a cavity) or from a region of interest interactively specified by the user (e.g, mouse cursor location). Our LOD visual abstraction consists of two distinct categories, a geometric abstraction and a shading abstraction.

The first category is the visual abstraction of geometry. The most domain-relevant visual representation is the solvent-excluded surface. Based on this representation the molecular biologists can claim whether a specific binding site is accessible to a solvent or not. The intermediate visual abstraction level is based on Gaussian kernel representation that approximates the SES and is often used in analysis of molecular surfaces despite of its lower expressive value with respect to the binding sites [Krone et al. 2011]. This visual abstraction is a compromise between the rendering performance and the expressiveness. The third, and the last level of the proposed visual abstraction scheme, is space-filling where individual atoms are represented by spheres. This is the fastest representation to render, however, its main usefulness is in providing a more gross structural context rather than providing a useful information about a local molecular detail (Fig. 3).

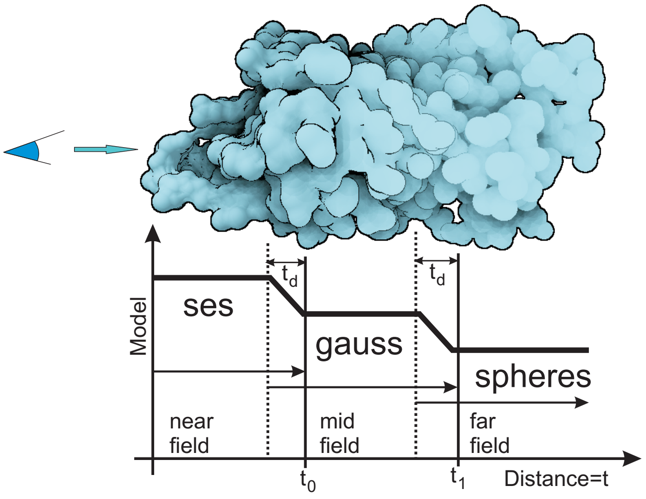

Figure 3: The organization of the three geometric and shading levels according to importance function $t(\mathbf{p})$ defined by the increasing distance from the camera. In the overlapping zones, the representations are merged using linear interpolation.

The second category is the shading visual abstraction. Together with geometry, we abstract from the details in shading in the following way. For conveying shape detail, we employ local diffuse shading model. For conveying relative depth, ambient occlusion is employed. Ordinal depth cues are communicated with contour rendering and the figure-ground ambiguity is resolved with silhouette rendering. This scheme is motivated by the workflow that David Goodsell, an acknowledged molecular scientist and illustrator, employs in molecular illustrations [Goodsell 2009]. In addition, we have added the detail level with local shading. While Goodsells illustrations have equal amount of visual cue for the entire molecular system, we have a specific distribution of visual cues for each level of detail. The figure-ground separation using silhouette and ambient occlusion as a relative depth cue are used for all abstraction levels. The near and mid-field levels additionally convey structural occlusion with contour rendering as an ordinal depth cue. The near-field conveys the shape, therefore it uses the diffuse shading, while the other two levels are represented with a constant shading, abstracting from atomic details. An example incorporating all abstraction levels is shown in Figure 3. The overall molecular rendering is performed by means of a ray-casting method, where each ray is incrementally processed allowing us to evaluate corresponding molecular and shading models.

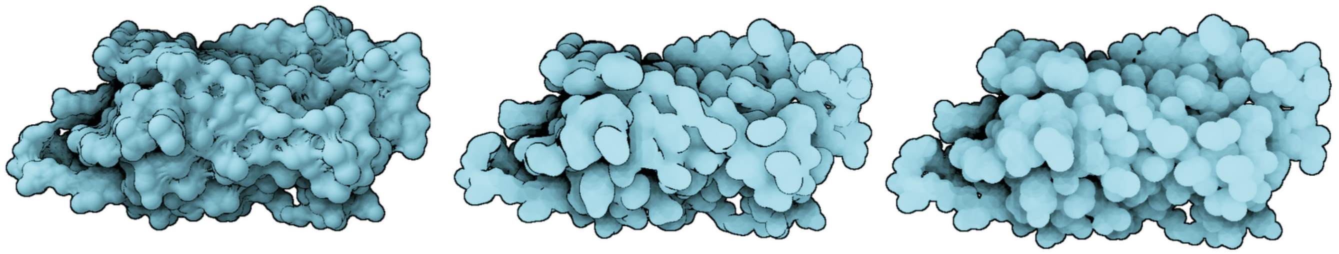

Figure 4: Three examples of water channel (Aquaporin) depicted using three different representations. Left: solvent excluded surface, Middle: Gaussian kernels, Right: van der Waals spheres.

4 Molecular Visual Abstraction

One of the important aspects that points out why we turn to implicit surfaces, is their ability to form a smooth transition or a blend between different implicit models easily. To give an example, when two implicit functions f and g overlap in space, the third function that defines a seamless transition between both of them in time, t, would be defined via a simple linear interpolation: $h = (1-t)f + tg$. This preserves the continuity of even two different representations, which is a very necessary property in order to achieve the seamless transition between different molecular models. It is important to mention that this property would be very hard to achieve with any boundary representation, especially on the real-time basis. An ad-hoc solution would be possible but not a general approach. It is also important to note that we propose abstraction levels that are aligned with visual processing. If there should be different levels, implicit representation can easily cope with them, while ad-hoc solutions cannot. In our work the interpolation parameter t is seen as an importance value $t = t(\mathbf{p})$ that is being varied in the scene, i.e, dependent on a given point $\mathbf{p}$ that is about the be evaluated. In our demonstrations we use the distance from the camera as the importance function, $t(\mathbf{p}) = ||\mathbf{eye} - \mathbf{p}||$. We will discuss the utilization of different choices of $t(\mathbf{p})$ in Section 6. We specify borders for all three areas (near, mid and far-field) using $t(\mathbf{p}) \leq t_0 \equiv$ near-field, $t_0 < t(\mathbf{p}) \leq t_1 \equiv$ mid-field and $t(\mathbf{p}) > t_1 \equiv$ far-field. Besides t_d represents the length of the transition area, which defines the blending interval between two distinct molecular surface representations. Thus when a point $\mathbf{p}$ lies in one area solely, we can evaluate a single implicit function, while for the overlapping areas we need to evaluate both functions and combine their result by linear interpolation.

To start let us assume that the set of atoms is defined as $C = \{(\mathbf{c}_1, r_1), \dots, (\mathbf{c}_n, r_n)\})$. Here we introduce the three implicit functions each defining the molecular model for one of the three intervals (Fig. 4).

Solvent Excluded Surface Representation: To represent SES by means of implicits, we take as a basis the approach proposed by Parulek and Viola [Parulek and Viola 2012]. The method for evaluating the implicit function has cubic complexity $O(n^3)$. This also represents one of the main reasons that we turn to the level-of-detail concept, so that we would be able to speed up the overall performance, while preserving SES model for the closest molecular parts from the camera. So far, there has not been any implicit method proposed to create SES on the fly faster than $O(n^3)$. Their method introduced the computation of SES using the solvent accessible surface (SAS). Essentially, SES representation is obtained by rolling a solvent represented by a ball of radius R, which subtracts the material from SAS. The method, mentioned above uses the computation of the closest point on the solvent accessible surface, $\mathbf{x}$, to a given point $\mathbf{p}$. Subsequently R is subtracted from the distance of those two points; $F_{ses} = ||\mathbf{x} - \mathbf{p}|| - R$. Even the formula looks very simple, to compute the point $\mathbf{x}$ one needs to test all possible triplets of atoms; i.e. therefore the cubic complexity. The final implicit function evaluates an exact Euclidean distance to the surface, although only to the distance R from the iso-surface of SES representation. One of the advantages of the proposed method is the flexibility of varying the parameters during rendering; e.g. atoms participating in SES representation, the solvent radius R. This represents the main reason to incorporate this method into our pipeline, which allows us to vary the length of the near-field easily.

Gaussian Kernels Representation: For the second, mid-field level, we utilize the Gaussian model. It smoothly blends the density field generated by the atoms, and also it forms a seamless transition between the SES and sphere models. The utilization of the Gaussian kernel for implicit modeling was used for the first time by Blinn [Blinn 1982]. He introduced the implicit function, describing the electron density function of atoms, by summing the contribution from each atom as follows: $F_{gauss}(\mathbf{p}) = T - \sum_i b_i e^{-a_i d_i^2}$, where d_i represents the distance from $\mathbf{p}$ to the center of atom c_i, b_i represents the "blobbiness", a_i describes the atom radius and T defines the electron density threshold. We adopted Blinn's model and specify the parameters a_i and b_i as are described in his paper: $b_i = R^2, a_i = -\ln r_i^2 / 2b_i$ and $T = 0.5$.

van der Waals Spheres Representation: Let us define a set of implicit functions defined as $\{f_1, f_2, \dots, f_n\}$, where each $f_i(\mathbf{p}) = r_i - ||\mathbf{p} - \mathbf{c}_i||$ represents an atom $\mathbf{c}_i$ with the corresponding van der Waals radius r_i. The implicit function defining the union of spheres, can be written as $F_{spheres}(\mathbf{p}) = \max\{f_1(\mathbf{p}), f_2(\mathbf{p}), \dots, f_n\mathbf{p}\}$, where the maximum operator represents the union term [Ricci 1972]. In order to render the iso-surface of $F_{spheres}$ solely, we actually do not need to evaluate the intersection of the ray and the function by a root finding method. Rendering can be efficiently solved by using ray-casting the spheres directly and storing just the closest depth values to the camera using OpenGL depth buffer. Therefore, even the function evaluation has still $O(n)$ complexity, the entire rendering pipeline can be optimized by drawing all the spheres in parallel, while the atomic operations evaluate the depth buffer operation. Moreover, the rendering process can be speed up by utilizing the sphere billboard technique [Daae Lampe et al. 2007]. To form a smooth blend between spheres and Gauss representation, we only need to evaluate $F_{spheres}$ in the transition area $t(\mathbf{p}) \in [t_1 - t_d, t_1]$. Although for the interval, $t(\mathbf{p}) > t_1$, we employ the sphere billboarding.

4.1 Seamless Transition

We define the interpolation only inside transition zones, while in the remaining ones it is always only one function evaluated. Since the importance function $t(\mathbf{p})$ is defined as a distance measure, its functional domain is in the interval $[0,\infty)$. To evaluate the implicit function F according to the field borders, we use the following branching scheme:

$$F = \begin{cases} F_{ses} & t(\mathbf{p}) \in [0, t_0 - t_d] \\ w_0 F_{gauss} + (1 - w_0) F_{ses} & t(\mathbf{p}) \in [t_0 - t_d, t_0] \\ F_{gauss} & t(\mathbf{p}) \in [t_0, t_1 - t_d] \\ w_1 F_{spheres} + (1 - w_1) F_{gauss} & t(\mathbf{p}) \in [t_1 - t_d, t_1] \\ F_{spheres} & t(\mathbf{p}) \in [t_1, \infty) \end{cases}, \quad (1)$$

where $w_{0,1}$ represents the parameter of linear interpolation, i.e., $w_{0,1} = (t_{0,1} - t(\mathbf{p}))/t_d$. We should emphasize that all three level-of-detail areas and their lengths can be specified interactively in real-time.

Here we would like to note the exploitation of linear interpolation instead of more sophisticated solutions, e.g., using variational methods [Turk and O'Brien 1999] or extended space mapping [Savchenko and Pasko 1998]. The both techniques provide several parameters to fine-tune the shape of the final interpolation. The drawback is that the both techniques are quite computational expensive and not suitable enough for real-time rendering applications. Therefore we turn to linear interpolation that represents the most simplest approach, which additionally fulfils our need for seamless transformation.

4.2 Shading Levels of Detail

Our shading model employs a set of visual abstractions that selectively enhance shape and depth information. The entire shading scheme is inspired by the approach presented by David Goodsell [Goodsell 2009]. We use his system of visual cues, i.e, constant shading, contour and depth enhancement, which he employs in molecular illustrations, although applied on spheres representation solely. We apply these visual cues in the focus and context style, where the focus is represented for the interval $t(\mathbf{p}) < t_0$. Here, we discuss the application of the aforementioned visual cues according all three level-of-detail areas.

In near-field $t(\mathbf{p}) \in [0, t_0]$, we employ a local diffuse shading model (DM), in combination with the constant shading model (CM), that is applied in accordance with $t(\mathbf{p})$ value. This enables us to create much smoother transitions to CM. In the translation zone $t(\mathbf{p}) \in (t_0 - t_d, t_0]$, we interpolate the shading model, for which DM continuously disappears towards the end of the translation area.

In the mid-field and far-field zones, $t(\mathbf{p}) \in [t_0, \infty)$, we employ just the constant shading model. The reason for applying CM for the mid-field is that Gaussian model conveys a less accurate solvent shape than SES. Thus using CM we are able to visually decrease surface discrepancies between both models (Fig. 1). Besides the shading, we incorporate silhouettes and contours into our visualization. There are several papers on contour enhancement techniques [Kindlmann et al. 2003]. We turn to curvature based techniques, which can suppress contours in low-curvature regions. On the other hand, those techniques are usually computationally demanding. Therefore, we adopt a technique introduced by Krüger et al. [Kruger et al. 2006], which approximates the view-dependent curvature by evaluation of two consequential gradients along the viewing ray. The *contour* predicate is then defined as follows:

$$\mathit{contour} \equiv \overrightarrow{\mathit{ray}} \cdot \nabla F_i > (C_t F_c \sqrt{2 - C_t F_c}), \quad (2)$$

where $F_c = \nabla F_i \cdot \nabla F_{i-1}$ represents a curvature approximated by two consequently evaluated gradients of function F at the i-th step. Parameter C_t reflects the contour thickness, which can be interactively specified by the user (Fig. 5). Furthermore, we preserve the contour for near-,and mid-field and neglect in the far-field. The reason behind discarding the contours in the context area defined by spheres is that they do not fully emphasize the inter-spherical space, i.e., just enhancing the spherical shape. In the second transition area, we scale the *contour* predicate to make the contour disappear continuously. To summarize, the shading model is evaluated as follows:

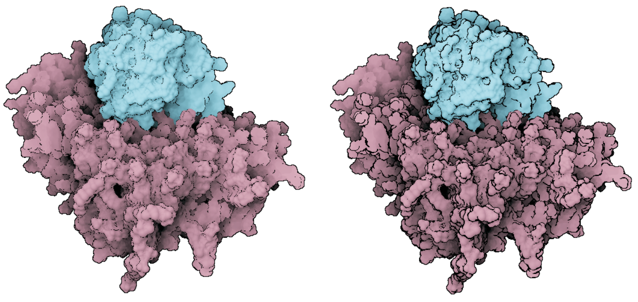

Figure 5: An example of changing the width of contours addressed by parameter C_t. For demonstration we use the phospholipase bound the lipid membrane. Left: $C_t = 0.05$, Right: $C_t = 0.4$.

$$C = \begin{cases} \mathit{contour\&DM} & t(\mathbf{p}) \in [0, t_0 - t_d] \\ \mathit{contour\&}(w_0 \mathit{CM} + (1 - w_0)\mathit{DM}) & t(\mathbf{p}) \in [t_0 - t_d, t_0] \\ \mathit{contour\&CM} & t(\mathbf{p}) \in [t_0, t_1 - t_d] \\ (1 - w_1) * \mathit{contour\&CM} & t(\mathbf{p}) \in [t_1 - t_d, t_1] \\ \mathit{CM} & t(\mathbf{p}) \in [t_1, \infty) \end{cases}, \quad (3)$$

where $w_{0,1}$ is defined in the same manner as in Eq. 1.

The silhouettes are generated with respect to the background of the rendered molecule, i.e., all the pixels that do not belong to the molecule are considered background. Afterwards, in the image space we perform edge detection to the binary texture where 1 represents molecule and 0 background. The silhouette is preserved for all three zones. This was chosen to imitate the Goodsell's approach and, additionally, to enhance the overall shape of the molecule. As the last step in our rendering pipeline, we add screen space ambient occlusion based on the method proposed by Luft et al. [Luft et al. 2006]. This, similarly to the silhouettes, is applied to all three zones.

5 Rendering and Performance Analysis

Our rendering pipeline consists of several steps. In the first one, we render the van der Waals atoms as spheres with an increased radius that defines their area of influence. This area is defined by means of solvent diameter $2R$, i.e., each atom is rendered as a sphere with its van der Waals radius increased by $2R$. The reasoning why to choose the solvent diameter as an area of the atom influence is described by Varshney et al. [Varshney et al. 1994]. Moreover, we do not perform sphere ray-casting, but instead quickly splat spheres using billboarding [Tarini et al. 2006].

Instead of displaying these spheres, we store them in the so-called A-buffer. The theoretical framework describing the A-buffer was presented by Carpenter in 1984 [Carpenter 1984]. Our implementation utilizes the recent shader extension that allows to read and write data in the fragment shader simultaneously. Essentially, A-buffer is a linked list of fragments generated for every pixel separately using atomic operations on the GPU. We define one global atomic counter that serves as the head pointer to the linked list. This counter is increased by one, every time there is a new fragment being generated

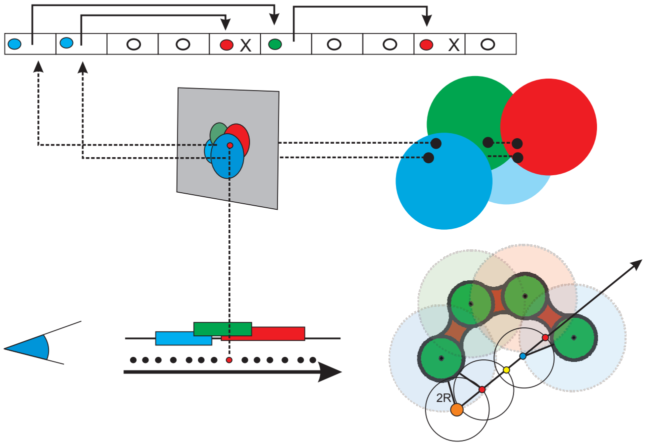

Figure 6: An illustration of the A-buffer. Top: Generating a linked list containing rendered spheres. Bottom: Stepping along the ray, where for each point we directly get a set of atoms that lie in the area of influence for a given point. Bottom-right: Ray-casting by means of sphere tracing. When a point (the yellow point) is in the area, where there is no atom of influence the point is automatically shifted to the first unprocessed atom along the ray (the blue point).

in the fragment shader. Each fragment record consists of the entry and the exit depth of a rendered atom, and the atom id. The fragment record is then stored in the shared image at the location addressed by the global counter (Fig. 6 top). It is important to mention that the similar approach for rendering molecules defined by blobby objects was presented by Szecsi and Illes [Szecsi and Illes 2012], which was called fragment linked list.

In the second step, before the actual ray-casting, we sort the fragment records increasingly according to the entry depth. This is a worthy investment, since when evaluating the ray-surface intersection test, it allows us to easily step along those atoms that participate to the actual point on the ray (Fig. 6 bottom left). It is important to mention that in [Szecsi and Illes 2012], it was assumed that the scene is already ordered. Here, the sorting is performed using CUDA, through the pixel buffer object compatibility, instead of utilizing the fragment shader, since we found that the performance has increased 4 times in our demonstrational scenarios. Thus for each image pixel (ray), we obtain a list of atoms that influence the function evaluation along the ray in ascending order (Fig. 6 bottom).

In the third step, the scene is rendered. Here the ray is cast for each image pixel, where we generate an input 3D point **p** based on the entry depth of the first sphere at the pixel location and the projection matrix (Fig. 6 the orange point, bottom right). Afterwards, we employ sphere tracing algorithm [Hart 1994] that processes the ray in a step-wise fashion until the last sphere exit depth is reached or we hit the iso-surface, i.e., $|F| \leq \varepsilon$. The selection of ε can be used to either increase the surface detail or to improve the rendering performance. When a point on the ray is in the area where no sphere of influence is presented, the point is automatically shifted to the first unprocessed sphere along the ray, i.e., the next one in the linked list (Fig. 6). This allows us to perform empty space skipping very efficiently.

Here we describe the performance analysis, where the lengths of individual fields across the molecule are varied. We show that the user has the possibility to alter the fields to either get more molecular details with decreased FPSs or other way around.

Since our framework introduces several principal parameters, it is quite challenging to evaluate the overall performance with regards to all of them. The possible combinations include varying lengths of all three fields, the length of the transition area and also the iso-surface precision parameter ε. We introduce evaluation based on several examples of molecules of various sizes, where we alter the lengths of near, mid and far-field while having fixed size of the transition area as well as the precision parameter. We setup $t_d = 4R$ and $\varepsilon = 0.05R$, where R is the solvent radius. The performance measurements are done on a workstation equipped with two (2 GHz) processors and 12.0 GB RAM and with the GPU, NVIDIA GeForce GTX 690.

It is important to mention here that for each frame we perform all the steps presented in Section 5, i.e., all three molecular visual representations are computed on the fly. One of the biggest advantages on the real-time based implicit function evaluation is the possibility of varying the function parameters anywhere in space, while preserving the interactive system response, which can show its potential in the future. To generate a suitable description of the performance based on the lengths of three fields, we store all FPS values for each distribution of fields. Afterwards, we employ ternary plots displaying a coverage of the three areas in barycentric coordinates. The colors, from yellow to red, encode the achieved FPS. For simplicity, we use relative length of fields expressed in percentage of how much of the molecule participates to each field; e.g.; $t_0 = 1/3$ and $t_1 = 2/3$ represents equally distributed fields over the molecule, which is represented by the central point in all four plots. This evaluation method is applied to four molecules (Fig. 7), Aquaporin (1852 atoms) (a), proliferatic cell nuclear antigen (12555 atoms) (b), phospholipase bound the lipid membrane (34490 atoms) (c), asymmetric chaperonin complex (58674 atoms) (d).

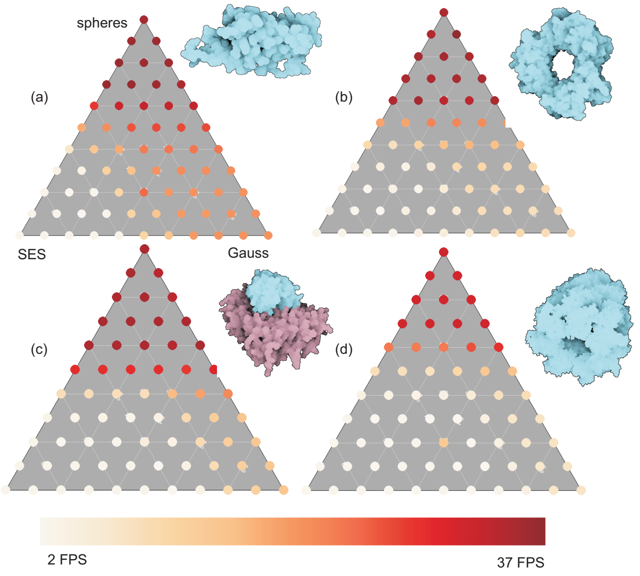

Figure 7: Ternary plots showing performance analysis evaluated on four distinct MD datasets. The analysis is based on the lengths of individual fields (spheres — near-field, Gauss — mid-field and SES — far-field). (a) Water channel (Aquaporin). (b) Proliferatic cell nuclear antigen. (c) Phospholipase bound the lipid membrane. (d) Asymmetric chaperonin complex. Note that the achieved FPS are, in the case of the camera based importance function, directly proportional to the lengths of each areas; i.e, prolongation of the near-field leads to decreasing FPSs on the other side, contraction of the far-field increases FPSs.

6 Results and Extensions

We demonstrate our technique on several molecules of various sizes. We employ the Protein Data Bank (PDB) file format, which stores the molecular information and initial atom positions. The MD trajectories of the atoms are stored in the *DCD* file format that is a standard in the Visual Molecular Dynamics (VMD) tool [Humphrey et al. 1996]. In order to visualize a molecular surface, we need to upload the corresponding set of atoms to the GPU. Therefore, our system can be used as a tool for browsing through large temporal datasets.

A typical demonstration of out technique is when the lengths of fields vary over the molecule and we fix the fields boundaries t_0 and t_1 and perform interactive zoom in towards the molecular center (Fig. 8). We have communicated the results with biologists and a biological illustrator, where we acquired a feedback about the overall visual quality and possible extensions of the proposed technique. Firstly, the illustrator was pleased with the results and the originality of the method. On the other side he suggested to improve the contour rendering for the SES portion of the model. Here the main issue he raised was that the contours are a little bit jaggy. This is indeed truth, since the issue lies in the SES model itself, which has C^1 discontinuities on the iso-surface that emerge essentially from the model definition. Such discontinues areas are also hard to track via the sphere tracking algorithm, which we also employ for the contour predicate. Here we use two neighboring gradients on the ray, and since the step size varies this can cause the discontinuities on the contour as well. Although this does not represent the primary goal of the paper, it should be studied in the future.

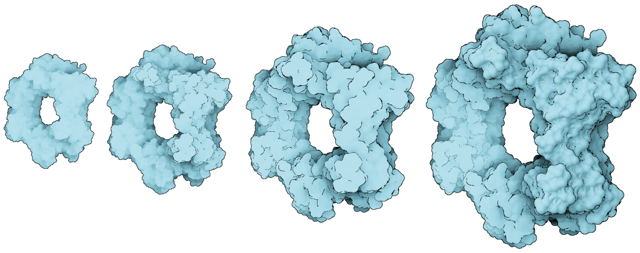

Figure 8: An example of zooming in towards the molecule (proliferatic cell nuclear antigen). When parameters t_0 and t_1 are fixed, we obtain more details at every zoom level.

Secondly, we were suggested to incorporate more silhouettes into the final visualization, which should be delineated between distinct molecules when we analyze compound systems. This note should be considered definitely for the future work. Domain experts found the achieved visuals original and helpful, mainly the interplay between the visualizations and the precision. Furthermore, they suggested to apply the proposed method to more application oriented scenarios. Therefore, we present two possible scenarios how this technique, especially the choice of the importance function $t(\mathbf{p})$, can be extended to more general focus and context applications in the domain of molecular analysis.

Mouse interaction. Firstly we introduce an interactive approach that allows users to interact with the importance function via the mouse handling directly. The importance function $t(\mathbf{p})$ is defined as a pixel-wise distance from the mouse cursor position. Again a user has the possibility to alter the lengths of fields. Such a method can be thought of as a magic-lens metaphor, which can seamlessly reveal more details on the molecules (Fig. 9 top).

Cavity-based abstraction. Secondly, we apply our method to enhance cavity or pockets visualization on molecular simulations. Here each cavity is represented by a central graph. The importance function $t(\mathbf{p})$ is defined as a minimal distance from the cavity graph (Fig. 9 bottom). A user can switch interactively between different graphs, where automatically the SES representation is shown only in a close neighborhood of the graph while smoothly disappearing into Gaussian and spheres models in the areas farther away from the graph skeleton.

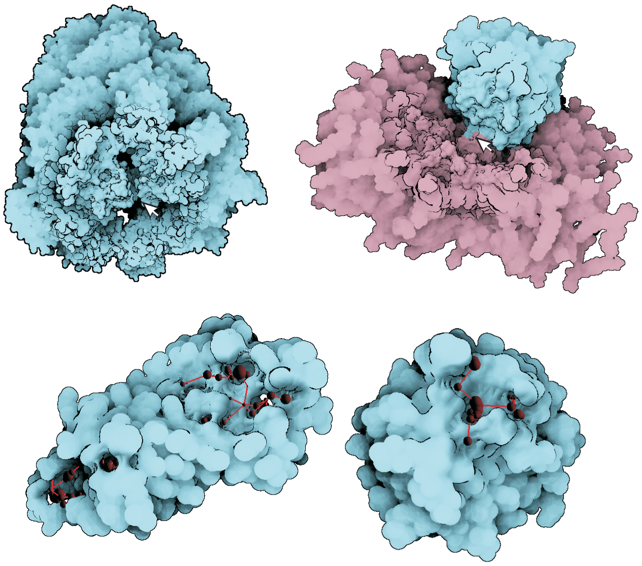

Figure 9: Top: Two LOD examples for mouse based interaction, where the distance from the mouse position determines individual fields. Left: An LOD example on Asymmetric chaperonin complex (1AON). Right: Revealing more details in the binding area between phospholipase and the lipid membrane. Bottom: An example of cavity-based abstraction on two proteins, where the distance from the extracted cavity centerlines determines individual fields.

To summarize the results, through our LOD concept we are able to boost the rendering performance of molecular models by $5-10\times$, while keeping the most detailed SES representation for the closest parts of the molecule from the camera. Besides all three representation are evaluated on-the-fly during ray-casting, which provides us with a great flexibility with regards to either enhancing the performance or the details for dynamic datasets.

7 Conclusion

We have proposed a novel approach for visualization of molecular surfaces. This enabled us to show even large protein complexes over time interactively. Our method utilizes the level-of-detail concept by means of three different molecular surface models, solvent excluded surface (SES), Gaussian kernels and van der Waals spheres combined in one visualization. Moreover, we introduced three shading levels that are aligned with the three surface models. For the realization, we took an inspiration from illustrations showing densely populated scenes with similar objects (spheres model with almost no detail), which are smoothly interconnected with highly detailed structures (SES model with full details) through the visual abstraction (Gaussian kernels model with fading out details). The importance function that represents the choice of the surface and shading models is based on the distance from the camera. We showcased how this can be effectively used to increase the rendering performance even for large molecules by interactive specification of level-of-detail boundaries. The entire rendering pipeline is performed on the single frame basis allowing to display any kind of molecular datasets outright. Although we have not experimented with the streaming from the simulations directly; nevertheless, this can be seen as a very suitable application scenario. Our LOD scheme fits nicely to the concept of the general focus and context visualization. We showed that the focus areas do not need to be specified through the camera position only, but also through different regions (mouse position) or objects (cavity graph centerlines) of interest. Also we introduced an LOD shading scheme with respect all three fields individually. We preserved seamless transition of depth, figure and shape visual cues using interpolation of shading and model schemes. A figure-ground ambiguity is solved via the utilization of the silhouette. The silhouette also keeps the entire molecule, even divided into distinct fields, perceptually unified.

Acknowledgements: We give thanks to Nathalie Reuter for providing the molecular dynamics simulation datasets, David Goodsell for giving us the necessary feedback for the overall visualization, and Barbora Tencerova and Cagatay Turkay for final touches with the paper. This work has been carried out within the PhysioIllustration research project (# 218023), which is funded by the Norwegian Research Council. This paper has been also supported by the Vienna Science and Technology Fund (WWTF) through project VRG11-010, and also by grants from the Excellence Center at Linköping and Lund in Information Technology (ELLIIT) and the Swedish e-Science Research Centre (SeRC), as well as VR grant 2011-4113.

References

ANDREWS, B. 2006. Introduction to "perceptual principles in medical illustration". In *ACM SIGGRAPH 2006 Courses*, ACM, New York, NY, USA, SIGGRAPH '06.

BLINN, J. 1982. A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1*, 235–256.

CARPENTER, L. 1984. The a -buffer, an antialiased hidden surface method. *SIGGRAPH Comput. Graph. 18*, 3 (Jan.), 103–108.

DAAE LAMPE, O., VIOLA, I., REUTER, N., AND HAUSER, H. 2007. Two-level approach to efficient visualization of protein dynamics. *IEEE transactions on visualization and computer graphics 13*, 6, 1616–23.

FALK, M., KRONE, M., AND ERTL, T. 2012. Atomistic visualization of mesoscopic whole-cell simulations. In *EG Workshop on Visual Computing for Biology and Medicine*.

FREY, S., SCHLOMER, T., GROTTEL, S., DACHSBACHER, C., DEUSSEN, O., AND ERTL, T. 2011. Loose capacity-constrained representatives for the qualitative visual analysis in molecular dynamics. In *IEEE Pacific Visualization Symposium*, 51–58.

GOODSELL, D. 2009. *The Machinery of Life*. Springer.

GROTTEL, S., REINA, G., DACHSBACHER, C., AND ERTL, T. 2010. Coherent culling and shading for large molecular dynamics visualization. *Comput. Graph. Forum 29*, 3, 953–962.

HART, J. C. 1994. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer 12*, 527–545.

HUMPHREY, W., DALKE, A., AND SCHULTEN, K. 1996. VMD: visual molecular dynamics. *Journal of molecular graphics 1*, 14, 33–38.

KINDLMANN, G., WHITAKER, R., TASDIZEN, T., AND MOLLER, T. 2003. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, IEEE Computer Society, Washington, DC, USA, VIS '03, 67–.

KRONE, M., FALK, M., AND REHM, S. 2011. Interactive Exploration of Protein Cavities. *Computer Graphics Forum 30*, 3, 673–682.

KRUGER, J., SCHNEIDER, J., AND WESTERMANN, R. 2006. Clearview: An interactive context preserving hotspot visualization technique. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (sept.-oct.), 941 –948.

LINDOW, N., BAUM, D., AND HEGE, H.-C. 2012. Interactive rendering of materials and biological structures on atomic and nanoscopic scale. *Computer Graphics Forum (accepted for publication) 31*, 3.

LUFT, T., COLDITZ, C., AND DEUSSEN, O. 2006. Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics 25*, 3 (jul), 1206–1213.

PARULEK, J., AND VIOLA, I. 2012. Implicit representation of molecular surfaces. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis 2012)*, 217–224.

RICCI, A. 1972. A constructive geometry for computer graphics. *The Computer Journal 16*, 2, 157–160.

RICHARDS, F. M. 1977. Areas, volumes, packing, and protein structure. *Annual Review of Biophysics and Bioengineering 6*, 1, 151–176.

SAVCHENKO, V., AND PASKO, A. 1998. Transformation of functionally defined shapes by extended space mappings. *The Visual Computer 14*, 5-6, 257–270.

SHARMA, A., KALIA, R. K., NAKANO, A., AND VASHISHTA, P. 2004. Scalable and portable visualization of large atomistic datasets. *Computer Physics Communications 163*, 1, 53–64.

SZECSI, L., AND ILLES, D. 2012. Real-Time Metaball Ray Casting with Fragment Lists. Eurographics Association, Cagliari, Sardinia, Italy, C. Andujar and E. Puppo, Eds., 93–96.

TARINI, M., CIGNONI, P., AND MONTANI, C. 2006. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics 12*, 5, 1237–1244.

TURK, G., AND O'BRIEN, J. F. 1999. Shape transformation using variational implicit functions. *Computer Graphics 33*, Annual Conference Series, 335–342.

VAN DER ZWAN, M., LUEKS, W., BEKKER, H., AND ISENBERG, T. 2011. Illustrative molecular visualization with continuous abstraction. *Computer Graphics Forum 30*, 3, 683–690.

VARSHNEY, A., BROOKS, JR., F. P., AND WRIGHT, W. V. 1994. Computing smooth molecular surfaces. *IEEE Comput. Graph. Appl. 14* (September), 19–25.

WEBER, J. R. 2009. ProteinShader: illustrative rendering of macromolecules.