DOI: 10.1111/cgf.12349

COMPUTER GRAPHICS *forum*
Volume 33 (2014), number 6 pp. 276–287

Continuous Levels-of-Detail and Visual Abstraction for Seamless Molecular Visualization

Julius Parulek[1], Daniel Jönsson[2], Timo Ropinski[2], Stefan Bruckner[1], Anders Ynnerman[2] and Ivan Viola[1,3]

[1]Department of Informatics, University of Bergen, Bergen, Norway
{julius.parulek, stefan.bruckner}@uib.no
[2]Department of Science and Technology, Linköping University, Linköping, Sweden
{daniel.jonsson, timo.ropinski, anders.ynnerman}@liu.se
[3]The Institute of Computer Graphics and Algorithms, Vienna University of Technology, Vienna, Austria
viola@cg.tuwien.ac.at

Abstract

Molecular visualization is often challenged with rendering of large molecular structures in real time. We introduce a novel approach that enables us to show even large protein complexes. Our method is based on the level-of-detail concept, where we exploit three different abstractions combined in one visualization. Firstly, molecular surface abstraction exploits three different surfaces, solvent-excluded surface (SES), Gaussian kernels and van der Waals spheres, combined as one surface by linear interpolation. Secondly, we introduce three shading abstraction levels and a method for creating seamless transitions between these representations. The SES representation with full shading and added contours stands in focus while on the other side a sphere representation of a cluster of atoms with constant shading and without contours provide the context. Thirdly, we propose a hierarchical abstraction based on a set of clusters formed on molecular atoms. All three abstraction models are driven by one importance function classifying the scene into the near-, mid- and far-field. 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.

Keywords: level of detail algorithms, implicit surfaces, clustering, scientific visualization

ACM CCS: Computer Applications [J.3]: Life and Medical Sciences Biology and Genetics COMPUTER GRAPHICS [I.3.3]: Picture/Image Generationa Viewing algorithms

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 data sets do no longer consist of only one moderately sized macromolecule, but instead of molecular systems representing complex interactions, for example, a phospholipid vesicle membrane together with proteins anchored in the membrane (Figure 1, right). One can easily obtain data sets where tens or hundreds of thousands of atoms are animated throughout a series of 1000 time-steps.

To analyse a binding site, a special visual representation is most popular among molecular biologists known as the solvent-excluded surface (SES) [Ric77]. 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 information provided by SES and investigate molecules with blobby Gauss kernel representations [Bli82], or with a simple space filling approach. The latter one, for example, can be represented very quickly by impostor-based sphere splatting, but it does not answer precisely whether a solvent can bind at a specific location to a macromolecule. Another research question is how to abstract the molecular surface further, beyond representing each single atom. An example of where such an abstraction would be essential is a *Powers-of-Ten* zooming interactive environment where the user can zoom-in onto a single atom, or zoom out to see a cellular-level structure of the same entity. At the cellular-level it is completely out of the question to render each atom of a molecule and a hierarchical abstraction would be needed. In the search for the appropriate solution we turn to the visual crafts for inspiration, which have been already successfully applied on molecular visualization [vdZLBI11].

© 2014 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.

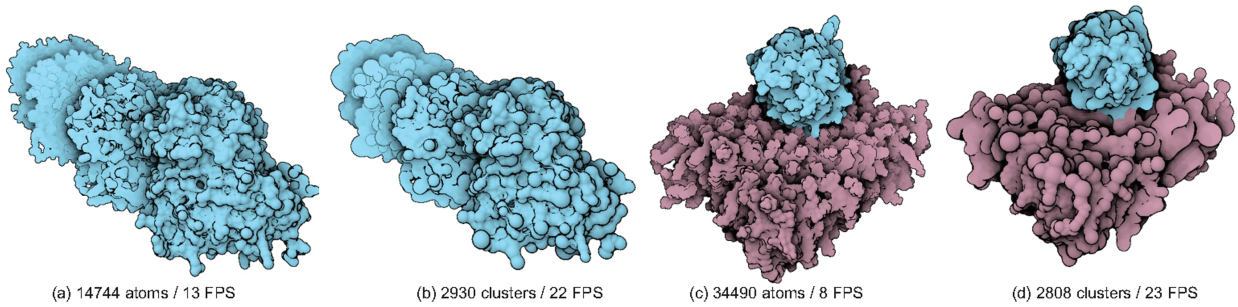

Figure 1: *Two molecular examples, Tubulin and phospholipase bound to lipid membrane, demonstrating utilization of our seamless visual abstraction. We employ three different surface representations [solvent-excluded surface (SES), Gaussian kernels and van der Waals spheres], their corresponding shading abstractions (diffuse shading and contours, constant shading with contours, constant shading without contours) and hierarchical representation. The application of individual levels is based on the distance to the camera; that is, the closest surface is based on highest surface, shading and hierarchical levels while the farthest are displayed via the lowest ones. In the presented examples we achieved* $5\times$ *to* $10\times$ *speed-up as compared to the full SES representation (a and c), and* $10\times$ *to* $20\times$ *when additionally applying hierarchical abstraction (b and d).*

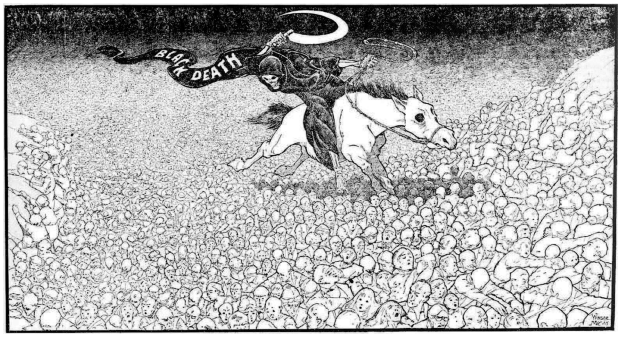

Figure 2: *Object constancy employed in visual arts by Winsor McCay 'When Black Death Rode'.*

Illustrators sometimes take a different approach to visually abstracting molecules from details. Instead of modifying the molecular representation into an entirely different molecular abstraction (such as transition between space filling representation and ribbons showing β-sheets and α-helices), they effectively use the perceptual principles of object constancy to depict structures that are too far away to recognize the details, through a simplified representation of that object. In this way the illustrators' manual creation process is speedup while at the same time also resulting in a more convenient visualization 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 Figure 2, which was exemplified by the professional scientific illustrator Bill Andrews [And06].

To address the molecular visualization challenge delineated above we propose to employ a seamless level-of-detail rendering scheme, in the same way as illustrators approach rendering of 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. We also utilize the level of detail scheme to guide the viewer to relevant information in the spirit of focus + context visualization techniques. The most detailed molecular surface representation is the SES representation, where every atom (except hydrogens) is rendered to form the molecular surface. Farther away from the viewer, we smoothly change the visual representation to an approximation of SES through Gaussian kernels. Structures farthest away from the view are represented by simple sphere splatting. When the individual atoms are no longer discernible, we employ hierarchical clusterings, where the atoms at a particular spatial location are grouped into *super*-atoms, which enables lower memory requirements and faster rendering performance. The use of these three levels of detail are motivated by the cognitive zones of the viewer: focus, focus-relevant and context zone (Figure 1). Generalizing the concept leads us to the definition of a 3-D importance function that can be based on the distance measure from a molecular feature, not only as a distance from the camera.

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 artefact that will involuntarily attract the attention of the biologist. To tackle this problem, we propose to utilize an implicit surface representation, where we can seamlessly blend from one surface representation to another one. 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 fulfills 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 can be summarized as follows: We propose a novel visualization approach that increases the overall rendering performance by utilizing a level-of-detail concept applied via hierarchical abstraction, surface abstraction and shading

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

abstraction. We build upon our earlier work [PRV13] on seamless molecular abstraction and extend it with respect to several aspects. Most notably, we present a method for hierarchical abstraction, which goes beyond the level of atomic detail. For this purpose we hierarchically cluster the entire molecular structure at various detail levels.

2. Related Work

Our approach builds on several aspects of previous work on molecular visualization, in particular with respect to choosing appropriate visual representations, methods for interactive rendering and level-of-detail techniques.

2.1. Visual representations

Tarini *et al.* present a real-time algorithm for visualizing molecules with the goal to improve depth perception [TCM06]. By combining ambient occlusion and edge-cueing together with graphics processing unit (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 [LVvdZ*11]. 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.* [vdZLBI11]. 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.* [TCM06], 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 [Web09]. In the work of Cipriano and Gleicher [CG07], spatio-physico-chemical properties are used to generate a simplified representation that conveys the overall shape. This approach, like many of the presented illustration models, goes back to the original work done by David Goodsell [Goo09], 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.* [FKE12], and it also inspired the creation of the renderings shown in this paper.

2.2. Interactive rendering

Besides recent efforts dealing with the visual representation of molecules, a lot of work has been dedicated to increase the overall rendering performance. For instance, Sharma *et al.* present an octree-based approach, which allows billions of atoms to be rendered interactively by exploiting view-frustum culling [SKNV04]. A combination of probabilistic and depth-based occlusion algorithms is used during rendering to determine the visible atoms. More recently, Grottel *et al.* have investigated different data simplification strategies which also incorporate culling [GRDE10]. 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 [DVRH07]. To represent these large-scale dynamic models, they also use a hierarchical approach where the topmost layer represents residues as 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 performed on-the-fly. Since SES represents the most advanced representation of molecular surfaces, which allows the molecule interactions and evolution to be studied, some effort has also been dedicated to improving the rendering of these fairly complex structures. Parulek and Viola propose an SES representation based on implicit surfaces [PV12]. By exploiting constructive solid geometry 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 pre-processing, they are able to vary SES parameters interactively. Frey *et al.* focus on MD simulation data [FSG*11]. In order to speed up rendering of these 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.

2.3. Level-of-detail techniques

Level-of-detail approaches have a long history in computer graphics [LWC*02]. Most techniques for rendering molecular data use surface simplification methods for generating different LODs [KOK06]. Lee *et al.* [LPK06] visualize large-scale molecular models using an adaptive level of detail (LOD) technique based on a bounding tree. Fraedrich *et al.* [FAW10] sample only the visible particles in the scene into perspective non-uniform grids in view space. These optimizations result in low computation times even for large data sets. They render the isosurfaces using GPU-based ray-casting. Krone *et al.* [KSES12] use a view-independent volumetric density map representation and generate a surface representation for rendering using a GPU implementation of the Marching Cubes algorithm similar to the work of Dias *et al.* [DG11]. The work of Bajaj *et al.* [BDST04] incorporates a biochemically sensitive level-of-detail hierarchy into the molecular representation and uses an image-based rendering approach. Although presented in the context of visual data mining in document collections, the H-BLOB method by Sprenger *et al.* [SBG00] is of relevance as it uses is a hierarchical clustering and visualization approach based on implicit surfaces. Our approach maintains an implicit representation throughout the pipeline and uses it for rendering directly. We use a hierarchical data representation scheme which forms, together with visual representation, and surface representation, a 3-D abstraction space. This provides us with fine-grained control over the different representational dimensions and enables us to flexibly

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

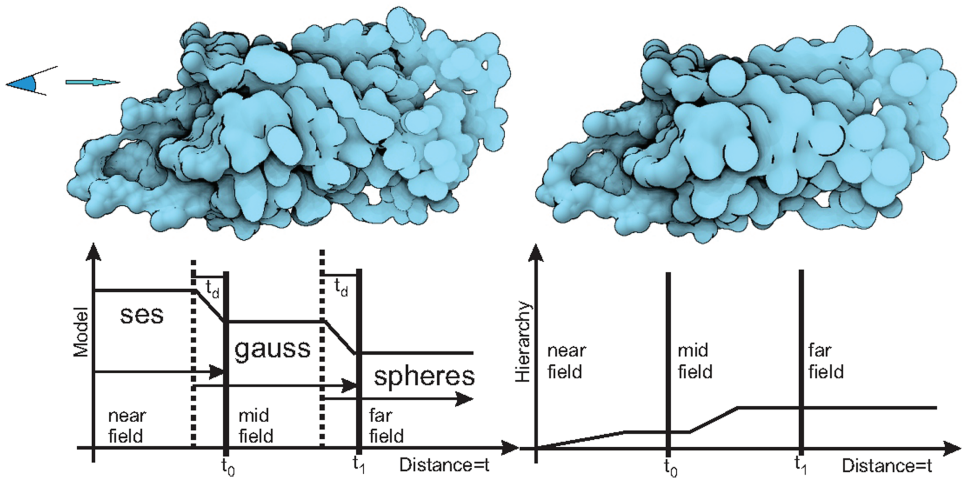

Figure 3: *Left: The organization of the three surface and shading levels according to importance function $t(\mathbf{p})$ defined by the increasing distance from the camera. In the overlapping zones, the representations are merged using linear interpolation. The molecule is displayed with the full atom count,* 1852 *atoms. Right: The extraction of cluster hierarchies based on $t(\mathbf{p})$. As the distance from the camera increases, the clusterings are retrieved from the higher hierarchical levels representing bigger clusters. An illustration shows exploitation of hierarchical abstraction containing* 784 *clusters, that is,* 42% *compress ratio.*

and seamlessly adjust the level of abstraction during interactive visualization.

3. Methodology

Motivated by the need for visualization of large molecular systems, we propose a seamless visual abstraction scheme which provides continuous transitions from the computationally expensive, but most relevant visualization technique, to the fastest representation which is suitable for representing the context. The key component of our approach which enables this seamless transition is an 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 interesting molecular feature (e.g. a cavity) or from a region of interest, interactively specified by the user (e.g mouse cursor location) [PRV13]. Our LOD visual abstraction consists of three distinct categories: hierarchical abstraction, surface abstraction and shading abstraction.

The first level of abstraction is concerned with whether the molecule is represented directly by atoms, or whether the atoms are grouped into clusters, that is, *superatoms*, which are then represented by a ball of a larger radius covering the volume of the grouped atoms (Figure 3, right). Our previous work [PRV13] included only the atomic level. Section 4.3 discusses the generation of this hierarchy.

The second category is the visual abstraction of surfaces. The most domain-relevant visual representation is the SES. Based on this representation the molecular biologists can decide whether a specific binding site is accessible to a solvent or not. The intermediate visual abstraction level is based on a Gaussian kernel representation that approximates the SES and is often used in analysis of molecular surfaces despite its lower expressive value with respect to the binding sites [KFR11]. This visual abstraction is a compromise between rendering performance and expressiveness. The last level of the proposed visual abstraction scheme is a space-filling approach 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 (Figure 3).

The third category is concerned with the visual abstraction of shading. Together with geometry, we abstract the details in shading in the following way. For conveying shape detail, we employ a local diffuse shading model. For conveying relative depth, ambient occlusion is used. Ordinal depth cues are communicated with contour rendering and the figure-ground ambiguity is resolved by silhouette rendering. This scheme is motivated by the workflow that David Goodsell, an acknowledged molecular scientist and illustrator, employs in molecular illustrations [Goo09]. We additionally provide a detail level with local shading. While Goodsell's illustrations have equal amount of visual cues for the entire molecular system, we have a specific distribution of visual cues for each level of detail. The figure-ground separation, which uses silhouette and ambient occlusion as a relative depth cue, is 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 and therefore uses 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

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

processed, thereby allowing us to evaluate corresponding molecular and shading models.

4. Molecular Visual Abstraction

The main reason for choosing an implicit representation is that it enables us to easily form a smooth transition, or a blend, between different types of surfaces. For instance, when two implicit functions f and g overlap in space, a simple way to generate a seamless transition between them is via linear interpolation: $h = (1-t)f + tg$. This preserves the continuity of even two different representations, which is a necessary property in order to achieve a seamless transition between different molecular models. This property would be very hard to achieve with any boundary representation, especially on a real-time basis. We propose a set of abstraction levels which are aligned with visual processing, but our framework can also easily handle additional levels. In our work the interpolation parameter t is interpreted as an importance value $t = t(\mathbf{p})$ that varies with the position $\mathbf{p}$ in the scene. In our demonstrations, we use the distance from the camera as the importance function, $t(\mathbf{p}) = ||\mathbf{eye} - \mathbf{p}||$. 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. The length of the transition area is controlled by t_d, which defines the blending interval between two distinct molecular surface representations. Thus, when a point $\mathbf{p}$ lies only in one area 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.

4.1. Surface abstraction

We assume a set of atoms defined as $C = \{(\mathbf{c}_1, r_1), \ldots, (\mathbf{c}_n, r_n)\}$ and introduce the three implicit functions each defining the molecular model for one of the three intervals.

4.1.1. *SES representation*

To represent SES by the means of implicits, we take as a basis the approach proposed by Parulek and Viola [PV12]. The method for evaluating the implicit function has cubic complexity $O(n^3)$. 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, for example, atoms participating in SES representation or the solvent radius R. This is the main reason why this method is incorporated into our pipeline; it enables us to vary the length of the near-field easily. This representation is the one that is most computationally expensive and it makes sense to apply it only when studying inter-atomic cavities in detail. Therefore, although in principle applicable to all hierarchical abstraction levels, it is only meaningful to utilize it in the near-field focal region.

4.1.2. *Gaussian kernel representation*

For the second, mid-field level of surface abstraction, we utilize the Gaussian model, which is widely used as an approximation of the SES model. It smoothly blends the density field generated by the atoms and also forms a seamless transition between the SES and sphere models. The utilization of the Gaussian kernel for implicit modelling was used for the first time by Blinn [Bli82] to describe 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 centre 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 specified the parameters a_i and b_i as were introduced by Grant and Pickup [GP95]: $b_i = R^2$, $a_i = -\ln r_i^2 / 2b_i$ and $T = 0.5$.

4.1.3. *van der Waals sphere representation*

Let us define a set of implicit functions defined as $\{f_1, f_2, \ldots, f_n\}$, 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 $F_{spheres}(\mathbf{p}) = \max\{f_1(\mathbf{p}), f_2(\mathbf{p}), \ldots, f_n\mathbf{p}\}$, where the maximum operator represents the union term [Ric72]. 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. Instead, rendering can efficiently be performed by ray-casting the spheres directly and storing the closest depth values to the camera in a depth buffer. Therefore, even if the function evaluation still has $O(n)$ complexity, the entire rendering pipeline can be optimized by drawing all the spheres in parallel, while the atomic operations evaluate the depth buffer. Moreover, the rendering performance can be increased by utilizing the sphere billboard technique [DVRH07]. To form a smooth blend between the van der Waals spheres and the Gaussian kernels representation we only need to evaluate $F_{spheres}$ in the transition area $t(\mathbf{p}) \in [t_1 - t_d, t_1]$. The sphere billboarding technique is employed when $t(\mathbf{p}) > t_1$.

We utilize linear interpolation between the representations inside transition zones, while the remaining zones only require a single representation to be evaluated. Our approach allows all three level-of-detail areas and their lengths to be modified in real time. We choose linear interpolation as it represents a simple, intuitive and efficient solution. More sophisticated approaches, for example, variational methods [TO99] or extended space mapping [SP98] provide several parameters to fine-tune the shape of the final interpolation, but are relatively expensive to evaluate and not yet suitable for real-time rendering applications.

4.2. Shading abstraction

Our shading model employs a set of visual abstractions that selectively enhance shape and depth information. The shading scheme is inspired by the approach presented by David Goodsell's artwork [Goo09]. We use his system of visual cues, that is, constant shading, contour and depth enhancement, which he employs in molecular illustrations, although applied only on our sphere representation. We apply these visual cues in a focus and context manner, where the focus is represented for the interval $t(\mathbf{p}) < t_0$. In the remainder of this section, we discuss the application of the aforementioned visual cues according to all three level-of-detail areas.

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

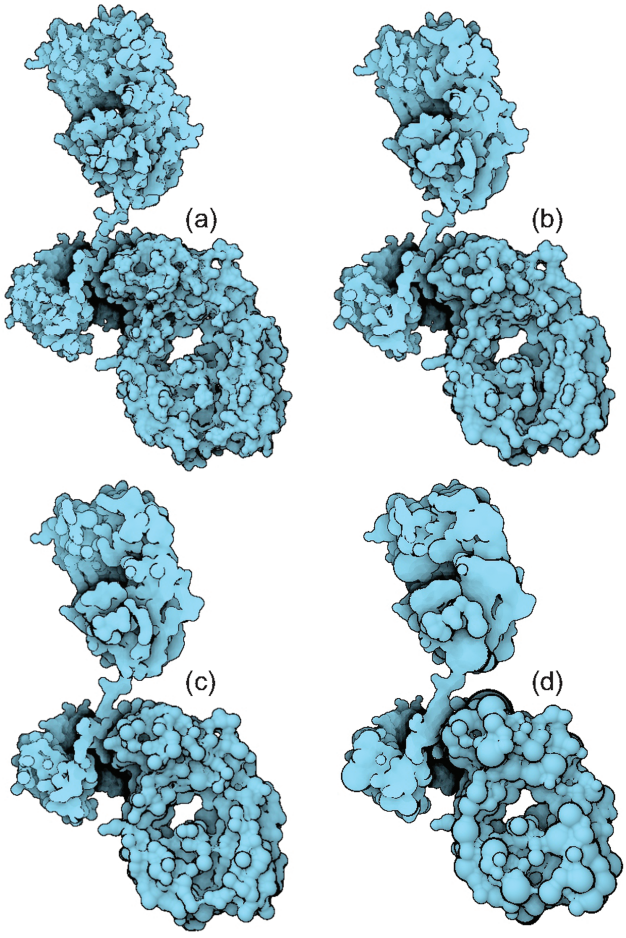

Figure 4: *Four hierarchical levels on immunoglobulin. The first level represented by the full atom count (a),* 12 530 *atoms. The second level (b), approximates the atoms in the first level by a set of* 6990 *clusters, which represents* 55.7% *elements of the full atom count. The other levels, (c) and (d), contain* 3122 (24.9%) *and* 1576 (12.5%) *elements.*

In the 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), which is applied in accordance with the $t(\mathbf{p})$ value. This enables us to create much smoother transitions to CM. In the transition zone $t(\mathbf{p}) \in (t_0 - t_d, t_0]$, we interpolate the shading model such that the DM continuously disappears towards the end of the transition area.

In the mid-field and far-field zones, $t(\mathbf{p}) \in [t_0, \infty)$, we employ constant shading model. The reason for applying the CM in the mid-field is that the Gaussian model conveys lower accuracy for the solvent shape than the SES. Thus, by using CM we are able to visually decrease the surface discrepancies between the two models (Figure 1). Besides the shading we incorporate silhouettes and contours into our visualization. We employ the approach of Kindlmann *et al.* [KWTM03] to generate contours of uniform thickness using the fast view-dependent curvature approximation of Krüger *et al.* [KSW06]. Furthermore, we preserve the contours

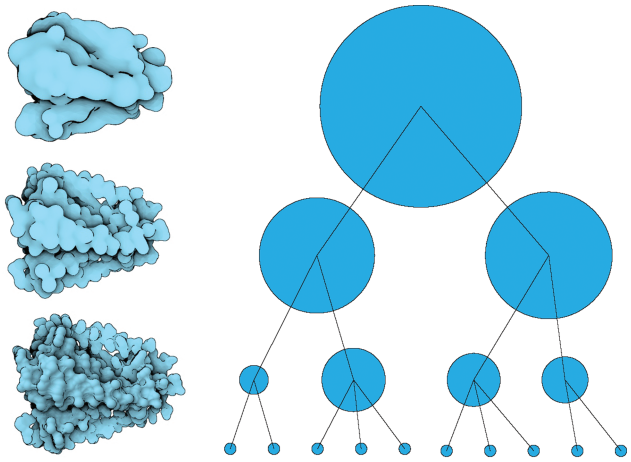

Figure 5: *A hierarchy based on spatial clustering is created using a bottom-up approach. The hierarchy is combined with the surface and shading abstractions using a seamless transition model.*

for near- and mid-field but neglect it 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, that is, just enhancing the spherical shape. In the second transition area, we scale the *contour* predicate to make the contour disappear continuously.

The silhouettes are generated with respect to the background of the rendered molecule, that is, all the pixels that do not belong to the molecule are considered background. Afterwards, in image space, we perform edge detection on the binary texture where 1 represents molecule and 0 means 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.* [LCD06]. The ambient occlusion is, similarly to the silhouettes, applied to all three zones.

4.3. Hierarchical abstraction

The visual representations discussed so far are based on access to the original resolution of the whole data set represented by individual atoms. Unfortunately, this limits the visualization to data sets that can fit and be streamed into memory in a timely manner. These representations are necessary when a structure is explored in detail, that is, by being able to go down to the atom level. While this is often desirable for the structures being directly in focus, or near the focus field, structures being far outside the focus do not need to convey this detailed information. Still, it is important that the overall structure of the molecule is conveyed and that large-scale features are preserved. Moreover, since state-of-the-art molecular visualization techniques [KBE09, LBPH10, PV12] exploit almost instant surface generation from the initial set of atoms prior to rendering the surface, it is necessary to enhance the rendering of large molecular and even cellular scenes using a new data simplification scheme as well.

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

To achieve this new level of abstraction, we again drew inspiration from David Goodsell's representations. By condensing the visualized information to the most essential visual elements, he is able to communicate the important features without introducing additional clutter. As the dominant visual elements in his representations are silhouettes and surfaces having a flat appearance, we have designed our hierarchical abstraction such that we can reduce a molecule to these elements. One alternative to achieve such an abstraction would be to use primary or secondary structures analysis. However, this would not generalize to other kinds of molecules, for example, lipids. Therefore, we propose to exploit location-based clustering, which enables support for a wider spectrum of molecules while generating a hierarchy of nested surface representations. By using a location-based clustering these nested surface representations will have a high degree of conservation with respect to the outline of the molecule (Figure 4). In the following, the initial set of atoms, $C = \{(\mathbf{c}_1, r_1), \ldots, (\mathbf{c}_n, r_n)\}$), becomes a set of spherical elements. Such an element can be either an atom or a cluster described again by a centre $\mathbf{c}_i$ and a radius r_i. More importantly, the function evaluation procedure is the same for all three surface representations. While the presented approach is generalizable, it can also be integrated with the implicit surface abstractions in order to create a seamless visual transition for the user.

In order to form the hierarchical representation of the particles, we use a bottom-up approach. The hierarchy generation process therefore starts with performing a spatial clustering on the original atom data. After the clustering is complete each formed cluster is represented by a sphere with a radius r, which bounds all the particles within the cluster, and the cluster centre c, which is the centre of gravity computed from cluster members. The next level of detail is created using the clusters from the previous level of detail as input and raising the error threshold by a factor of two. This process continues until a maximum number of levels of details have been created or only a single cluster remains. Each particle/cluster will therefore have a single parent, but can have multiple children as illustrated in Figure 5.

Several clustering methods have been studied, where the computational complexity was considered crucial. We analysed the following clustering algorithms: DB-SCAN, k-means, hierarchical and the Affinity Propagation (AP) technique [Llo82, FD07, Mül13]. As a notable result, we found out that applying a density-based clustering scheme does not perform well on molecular data sets. The main reason is that molecular objects inherently lack any significant variation within the atom density distribution. Therefore, applying the DB-SCAN often produces only a single cluster for the entire molecule. Moreover, by testing various molecules, it became clear that qualitatively the AP algorithm performed best for the tested data sets. The AP algorithm was presented by Frey and Dueck [FD07], having formed clusters more uniformly and with a lower error bound than any other clustering algorithms. However, the biggest drawbacks of the AP algorithm is its computation complexity, $O(n^2)$. Indeed, AP is by far the slowest algorithm in our test group. Even though the cluster coverage between neighbouring levels is far better than using the remaining techniques, when the atom count is more than 10 K, creating already a single cluster level takes tens of minutes. Similarly, k-means also provides a computationally expensive solution, which prohibits its application on large molecules. On the other hand, the fast hierarchical clustering method, proposed by Müllner [Mül13], offers a very fast solution with a high-quality cluster coverage at the same time. A comparison of applying the three clustering algorithms is depicted in Figure 6. Generation of five hierarchical levels takes 30 923 (ms) for AP, 23 508 (ms) for k-means and 676 (ms) for hierarchical clustering.

5. Rendering and Performance Analysis

Our rendering pipeline consists of several steps (Figure 7). In the first one, we traverse the cluster hierarchy in top–down manner to retrieve all the clusters/atoms that are about to be used for the molecular representation and visualization. Starting from the highest level, we evaluate whether a cluster C is directly used in the visualization or whether it is required to recursively evaluate its child nodes (clusters or atoms in the leaves). The evaluation criterion that decides whether a cluster C is going to be added to the display list is defined by function

$$g(C, t) \equiv t > \frac{l_C}{l_{max}} t_1, \tag{1}$$

where l_C is the hierarchy level of C, l_{max} is the highest available hierarchy level and t_1 represents the far-field depth. When a cluster meets the criteria, it is added to the display list (Figure 8). The hierarchy traversal is performed on the CPU side, as a frame preprocessing, before the display list is sent to GPU. Here we have not found any performance drop even for larger hierarchical trees.

In the second step, we render clusters/atoms, stored in the display list, as spheres with an increased cluster radius that defines their area of influence. This area is defined by means of solvent diameter $2R$, that is, each cluster is rendered as a sphere with its cluster 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.* [VBW94]. Moreover, we do not perform sphere ray-casting, but instead quickly splat spheres using billboarding [TCM06].

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 [Car84]. 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 when a new fragment is generated in the fragment shader. Each fragment record consists of the entry and the exit depth of a rendered cluster, and the cluster id. The fragment record is then stored in the shared image at the location addressed by the global counter. It is noteworthy to mention that similar approaches for rendering molecules, defined by blobby objects and iterative blending, were presented by Szecsi and Illes [SI12] and Parulek and Brambilla [PB13], respectively.

In the third step, before the actual ray-casting, we sort the fragment records by entry depth. This allows us to easily step along those clusters during subsequent ray-casting. Sorting is performed using CUDA, as it proved to be substantially faster (more than a factor of 4) than a fragment shader implementation in our experiments. Thus

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

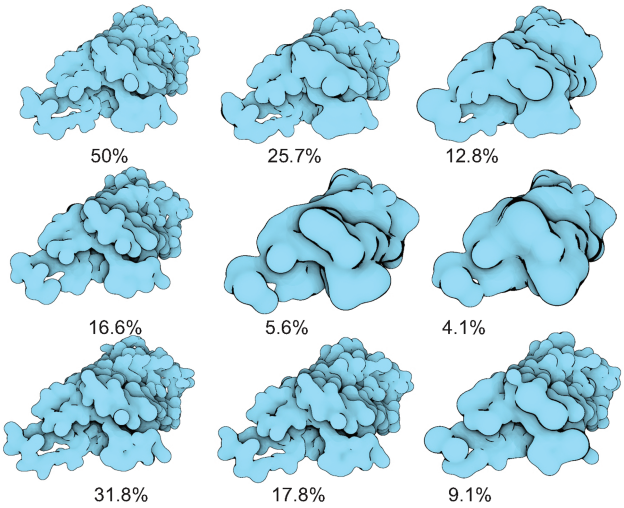

Figure 6: *Comparison of three clustering algorithms. The top row presents the first three levels of k-means clustering, the second row demonstrates affinity propagation method and the third row stands for fast hierarchical clustering. The percentage represents the ratio between the cluster render list size and the full atom count. Note that the ratios are different due to characteristics of the employed algorithms providing dissimilar clusters.*

for each image pixel (ray), we obtain a list of clusters that influence the function evaluation along the ray in ascending order.

In the fourth step, the scene is rendered. Here the ray is cast for each image pixel, where we generate an input 3-D point **p** based on the entry depth of the first sphere at the pixel location and the projection matrix. Afterwards, we employ a sphere tracing algorithm [Har94] that processes the ray in a stepwise fashion until the last sphere exit depth is reached or we hit the iso-surface, that is, $|F| \leq \epsilon$. 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, that is, the next one in the linked list. This allows us to perform empty space skipping very efficiently.

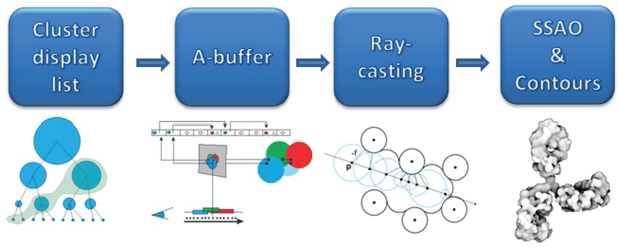

Figure 7: *An illustration of the rendering pipeline. The formation of the cluster display list is determined by Equation (1). Clusters are represented as spheres that are rendered into the A-buffer. The ray-casting is performed through sphere tracing algorithm. In the end, we compute screen space ambient occlusion.*

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 frames per second (FPSs) or vice versa.

While a comprehensive evaluation of the performance of our method with respect to all parameter combinations (varying lengths of all three fields, the length of the transition area and iso-surface precision parameter ϵ) is not feasible, we demonstrate its performance using several indicative examples. For the hierarchical abstraction, we utilize the fact that the performance is linearly dependent on the amount of clusters used. For instance, in Figure 1, when using just 20% (b) and 8% (d) of spheres compared to full atom count, (a) and (c), the performance increases almost $2\times$ and $3\times$. Therefore, we focus our performance analysis rather on the surface abstraction when using the full atom count. We introduce evaluation based on several examples of molecules of various sizes where we alter the lengths of near-, mid- and far-field, while choosing a fixed size for the transition area as well as the precision parameter. We setup $t_d = 4R$ and $\epsilon = 0.05R$, where R is the solvent radius. The performance measurements are performed 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 that for each frame we perform all the steps presented in Section 5. One of the biggest advantages of our real-time implicit function evaluation is the possibility of varying the function parameters anywhere in space, while preserving an interactive system response. 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 colours, 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;

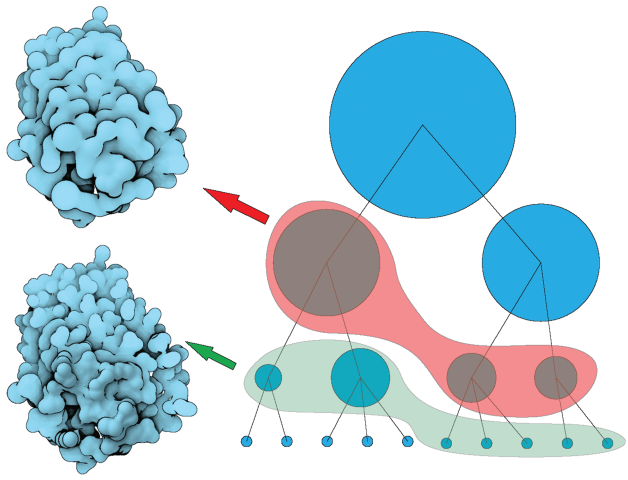

Figure 8: *An example of two display lists. During the cluster tree traversal, all the nodes that fulfill Equation (1), are added to the list. By zooming out outwards the molecule, the red display list becomes more reduced and abstracted than the green display list.*

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

for example, $t_0 = 1/3$ 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 (Figure 9), Aquaporin (1852 atoms) (a), proliferatic cell nuclear antigen (12 555 atoms) (b), phospholipase bound the lipid membrane (34 490 atoms) (c), asymmetric chaperonin complex (58 674 atoms) (d).

6. Results and Limitations

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 atom positions.

A typical demonstration of our 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 centre. Such an example is displayed in Figure 10 using the full atom count and also the hierarchical representation. Notice that on each zoom level there are some visual differences, but the higher cluster levels apply only when a molecule moves away from the viewer, where the visual discrepancies are even more suppressed. On the other side, in this example and for comparative purposes the non-clustered and clustered versions are depicted in the same size. In the figure, only 60% of spheres were employed for the rightmost visualization and 30% of spheres for the leftmost compared to the spheres/atoms contained within the molecule.

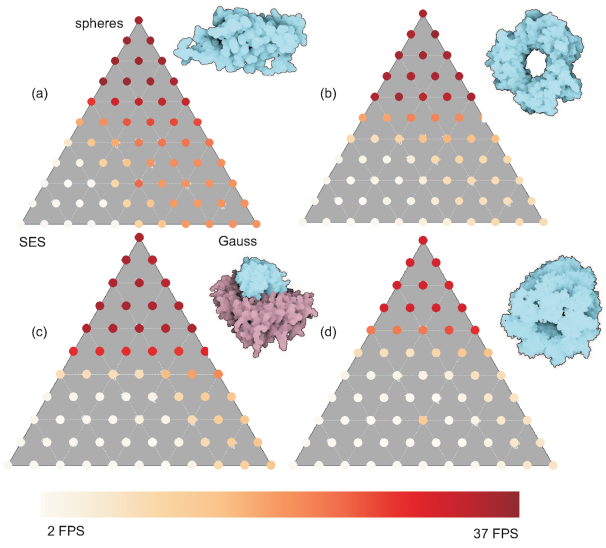

Figure 9: *Ternary plots showing performance analysis evaluated on four distinct molecular structures. The analysis is based on the lengths of individual fields (SES, near-field; Gauss, mid-field; spheres, far-field). (a) Water channel (Aquaporin). (b) Proliferatic cell nuclear antigen. (c) Phospholipase 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; that is, prolongation of the near-field leads to decreasing FPSs on the other side, contraction of the far-field increases FPSs.*

Through our LOD concept we are able to boost the rendering performance of molecular models by 5× to 10×, while keeping the most detailed SES representation for the closest parts of the molecule from the camera. Additionally, when applying the hierarchical representation, we get even up to 20× the frame rate compared to full SES representation. All three surface representations 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 data sets.

The utilization of hierarchical abstraction brings two major limitations. The first one is the actual surface precision when using the full atom count compared to exploiting the cluster hierarchy. Here, our shading abstraction helps to hide the most of the surface dissimilarities (Figure 10). Nevertheless, to compute the error quantitatively, we would need to firstly evaluate the most suitable parameters for the clustering method, for example, distance metric and stopping criteria, to reduce the error there first. The cluster error increases as we move up in the hierarchy. Nevertheless, the highest levels are usually employed only when depicting contextual molecular parts being farther away from the viewer.

The second limitation of utilizing the hierarchical abstraction is the requirement of performing the sequential clusterings. This has to be done for each new structure modification repetitively. For molecules containing a few thousands of atoms, formation of 5 to 10 hierarchical clusterings can take up to 1 s. While for larger molecules (molecular systems) this can take up to a minute. For example, forming five levels for the lipid–protein complex (Figure 1) took 20 s, while generation of five levels for asymmetric chaperonin complex (58 674 atoms) takes 80 s (Figure 11).

Nevertheless, we can already see the potential of our approach for visualizing mesoscopic whole-cell simulations [FKE12]. Here a cluster hierarchy can be formed in the pre-computation step for all acting molecules. Another potential solution to perform a clustering on dynamic structures, is to exploit a fast GPU-based bounding volume hierarchies (BVH). For instance, Bitner *et al.* introduce a GPU-based solution to update a BVH tree to minimize the overall cost function [BHH13], which in our case can be based on one of the abstraction levels.

We have demonstrated our method to biologists and a scientific 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 proposed concept. On the other hand, it was suggested to improve the contour rendering for the SES portion of the model. Here the main issue he raised was that the contours can appear jaggy which is due to C^1 discontinuities on the iso-surface of the SES model. Such discontinuous areas are also hard to track via the sphere tracing algorithm, which we also employ for the contour predicate. Additionally, the problem may be amplified by the fast curvature approximation we employ and a more costly scheme could help to overcome it. Overall, however, these issues were not seen as critical.

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

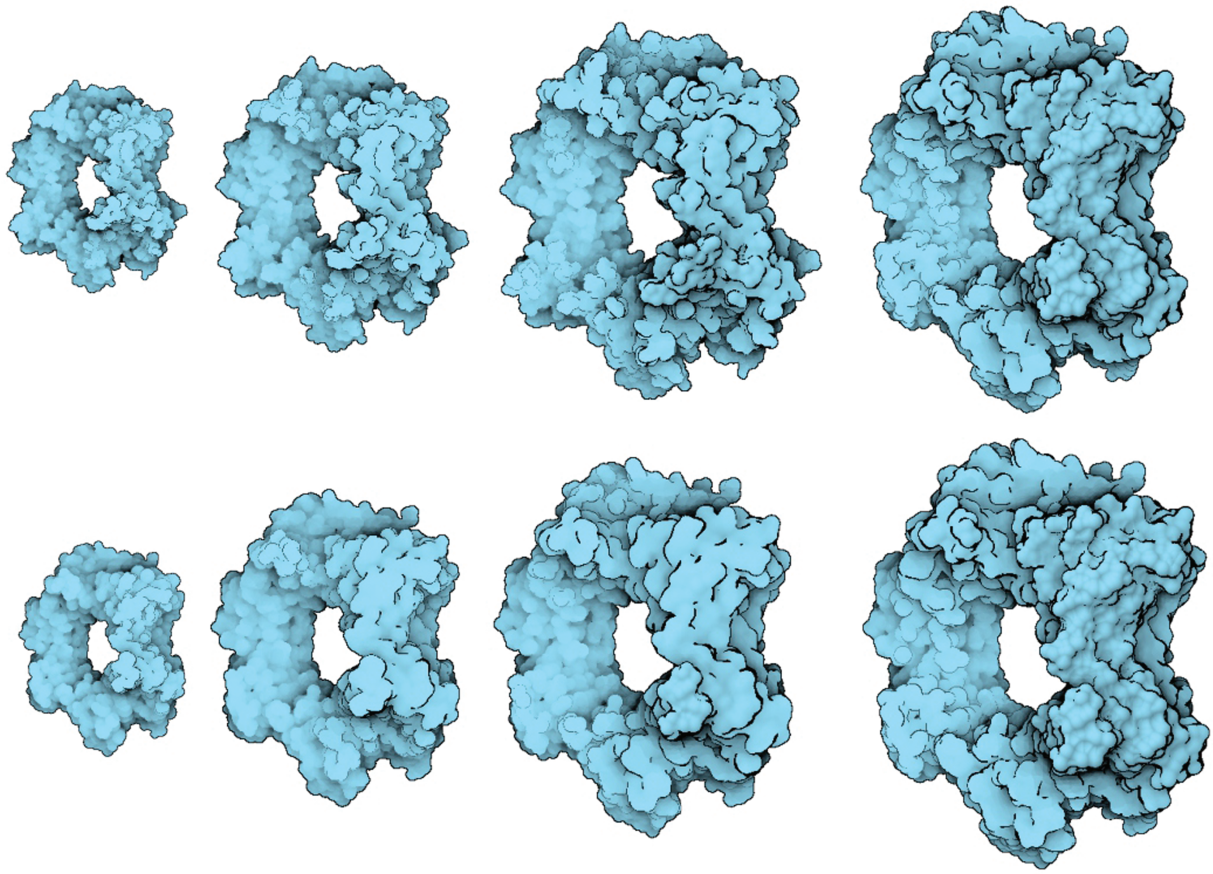

Figure 10: *Comparison of zooming in towards the molecule (proliferatic cell nuclear antigen) performed using the full atom count (*12 555 *atoms, top) and the hierarchical representation (bottom). The display list contains from left to right:* 3716*,* 4527*,* 5292 *and* 7516 *clusters.*

Furthermore, we were suggested to incorporate additional silhouettes into the final visualization to clearly delineate boundaries between distinct molecules in compound systems. While not the focus of this paper, we found that this is an important note to be considered in our future work. Domain experts found the achieved visuals original and helpful, mainly due to the interplay between the visualizations and the precision. Furthermore, they suggested to apply the proposed method to more application-oriented scenarios.

7. Summary

We have proposed a novel approach for visualization of molecular surfaces. Our approach is capable of rendering large protein complexes interactively, while rapidly reducing the amount of displayed primitives, and at the same, keeping the visual appearance similar to the original data. Our method utilizes the level-of-detail concept by means of three different molecular surface models, 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). Finally, we proposed a new hierarchical abstraction that approximates the molecular atoms with a set of clusters that are employed in the final visualization.

The importance function that represents the choice of the surface, shading and hierarchical models is based on the distance from the camera. We showcased how this can be used effectively 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-fly. We introduced an LOD shading scheme with respect to all three fields individually. We preserved a 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.

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

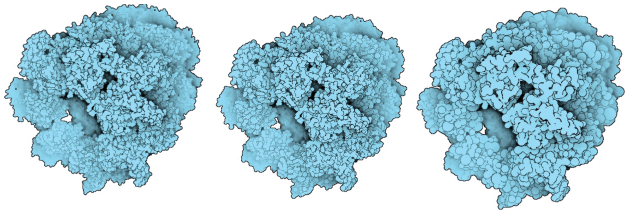

Figure 11: *An example of asymmetric chaperonin complex (*58 674 *atoms). Our representation can easily depict large molecules, where we employ the full atom count (left),* 36 740 *(middle) and* 9989 *(right) clusters.*

Acknowledgements

We thank Nathalie Reuter for providing the MD simulation data sets, David Goodsell and Helwig Hauser for giving us the necessary feedback for the overall visualization. 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 supported by the EC Marie Curie Career Integration Grant through project PCIG13-GA-2013-618680, and also by grants from the Excellence Center at Linköping and Lund in Information Technology (ELLIIT), the Swedish Research Council through the Linnaeus Center for Control, Autonomy, and Decisionmaking in Complex Systems (CADICS), and the Swedish e-Science Research Centre (SeRC), as well as VR grant 2011-4113.

References

[And06] Andrews B.: Introduction to 'perceptual principles in medical illustration'. In *ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), SIGGRAPH '06, ACM.

[BDST04] Bajaj C., Djeu P., Siddavanahalli V., Thane A.: Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. In *Proceedings of IEEE Visualization 2004* (Austin, TX, USA, 2004), pp. 243–250.

[BHH13] Bittner J., Hapala M., Havran V.: Fast insertion-based optimization of bounding volume hierarchies. *Computer Graphics Forum 32*, 1 (2013), 85–100.

[Bli82] Blinn J.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1* (1982), 235–256.

[Car84] Carpenter L.: The a -buffer, an antialiased hidden surface method. *SIGGRAPH Computer Graphics 18*, 3 (January 1984), 103–108.

[CG07] Cipriano G., Gleicher M.: Molecular surface abstraction. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1608–1615.

[DG11] Dias S. E., Gomes A. J.: Graphics processing unit-based triangulations of blinn molecular surfaces. *Concurrency and Computation: Practice and Experience 23*, 17 (2011), 2280–2291.

[DVRH07] Daae Lampe O., 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.

[FAW10] Fraedrich R., Auer S., Westermann R.: Efficient high-quality volume rendering of SPH data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1533–1540.

[FD07] Frey B. J., Dueck D.: Clustering by passing messages between data points. *Science 315* (2007), 972–976.

[FKE12] Falk M., Krone M., Ertl T.: Atomistic visualization of mesoscopic whole-cell simulations. In *EG Workshop on Visual Computing for Biology and Medicine* (Norrköping, Sweden, 2012).

[FSG*11] Frey S., Schlomer T., Grottel S., Dachsbacher C., Deussen O., Ertl T.: Loose capacity-constrained representatives for the qualitative visual analysis in molecular dynamics. In *IEEE Pacific Visualization Symposium* (Hong Kong, China, 2011), pp. 51–58.

[Goo09] Goodsell D.: *The Machinery of Life*. Springer, New York, 2009.

[GP95] Grant J. A., Pickup B. T.: A Gaussian description of molecular shape. *The Journal of Physical Chemistry 99*, 11 (March 1995), 3503–3510.

[GRDE10] Grottel S., Reina G., Dachsbacher C., Ertl T.: Coherent culling and shading for large molecular dynamics visualization. *Computer Graphics Forum 29*, 3 (2010), 953–962.

[Har94] Hart J. C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer 12* (1994), 527–545.

[KBE09] Krone M., Bidmon K., Ertl T.: Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1391–1398.

[KFR11] Krone M., Falk M., Rehm S.: Interactive exploration of protein cavities. *Computer Graphics Forum 30*, 3 (2011), 673–682.

[KOK06] Kanai T., Ohtake Y., Kase K.: Hierarchical error-driven approximation of implicit surfaces from polygonal meshes. In *Proceedings of the Eurographics Symposium on Geometry Processing 2006* (Sardinia, Italy, 2006), pp. 21–30.

[KSES12] Krone M., Stone J., Ertl T., Schulten K.: Fast visualization of Gaussian density surfaces for molecular dynamics and particle system trajectories. In *Proceedings of EuroVis 2012 Short Papers* (Vienna, Austria, 2012), pp. 67–71.

[KSW06] Kruger J., Schneider J., Westermann R.: Clearview: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (September–October 2006), 941–948.

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MOLLER T.: Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (Washington, DC, USA, 2003), VIS '03, IEEE Computer Society.

[LBPH10] LINDOW N., BAUM D., PROHASKA S., HEGE H.-C.: Accelerated visualization of dynamic molecular surfaces. *Computer Graphics Forum 29*, 3 (2010), 943–952.

[LCD06] LUFT T., COLDITZ C., DEUSSEN O.: Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics 25*, 3 (July 2006), 1206–1213.

[Llo82] LLOYD S.: Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137.

[LPK06] LEE J., PARK S., KIM J.-I.: View-dependent rendering of large-scale molecular models using level of detail. In *Proceedings of the International Conference on Hybrid Information Technology 2006* (Cheju Island, Korea, 2006), pp. 691–698.

[LVvdZ*11] LUEKS W., VIOLA I., VAN DER ZWAN M., BEKKER H., ISENBERG T.: Spatially continuous change of abstraction in molecular visualization. In *Abstracts of 1st IEEE Symposium on Biological Data Visualization* (BioVis 2011, Providence, RI, USA, 2011), M. Meyer and C. Nielsen (Eds.), IEEE Computer Society.

[LWC*02] LUEBKE D., WATSON B., COHEN J. D., REDDY M., VARSHNEY A.: *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York, NY, 2002.

[Mül13] MÜLLNER D.: fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software 53*, 9 (5 2013), 1–18.

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

[PRV13] PARULEK J., ROPINSKI T., VIOLA I.: Seamless abstraction of molecular surfaces. In *Proceedings of the 29th Spring Conference on Computer Graphics* (Comenius University, Bratislava, Slovakia, 2013), pp. 120–127.

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

[Ric72] RICCI A.: A constructive geometry for computer graphics. *The Computer Journal 16*, 2 (1972), 157–160.

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

[SBG00] SPRENGER T. C., BRUNELLA R., GROSS M.: H-blob: A hierarchical visual clustering method using implicit surfaces. In *Proceedings of IEEE Visualization 2000* (Salt Lake City, Utah, USA, 2000), pp. 61–68.

[SI12] SZECSI L., ILLES D.: *Real-Time Metaball Ray Casting with Fragment Lists* (2012). C. Andujar and E. Puppo (Eds.), Eurographics Association, Cagliari, Sardinia, Italy, pp. 93–96.

[SKNV04] SHARMA A., KALIA R. K., NAKANO A., VASHISHTA P.: Scalable and portable visualization of large atomistic datasets. *Computer Physics Communications 163*, 1 (2004), 53–64.

[SP98] SAVCHENKO V., PASKO A.: Transformation of functionally defined shapes by extended space mappings. *The Visual Computer 14*, 5–6 (1998), 257–270.

[TCM06] 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.

[TO99] TURK G., O'BRIEN J. F.: Shape transformation using variational implicit functions. *Computer Graphics 33*, Annual Conference Series (1999), 335–342.

[VBW94] VARSHNEY A., BROOKS JR, F. P., WRIGHT W. V.: Computing smooth molecular surfaces. *IEEE Computer Graphics Applications 14* (September1994), 19–25.

[vdZLBI11] VAN DER ZWAN M., LUEKS W., BEKKER H., ISENBERG T.: Illustrative molecular visualization with continuous abstraction. *Computer Graphics Forum 30*, 3 (2011), 683–690.

[Web09] WEBER J. R.: ProteinShader: Illustrative rendering of macromolecules. *BMC Structural Biology 9*, 1 (2009), 1–19.

© 2014 The Authors
Computer Graphics Forum published by John Wiley & Sons Ltd.