

Coverage-Based Opacity Estimation for Interactive Depth of Field in Molecular Visualization

Sathish Kottravel*

Martin Falk*

Erik Sundén*

Timo Ropinski†

* Interactive Visualization Group, Linköping University, Sweden

† Visual Computing Research Group, Ulm University, Germany

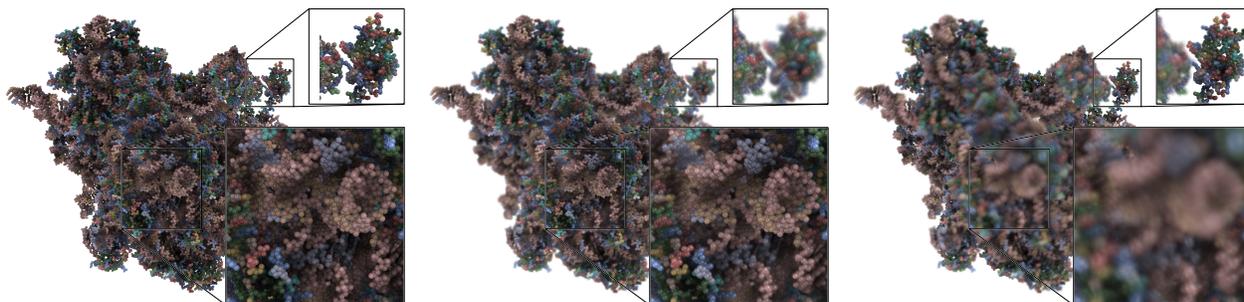


Figure 1: *Thermus thermophilus* 70S ribosome (PDB ID: 2WDK) rendered with no Depth of Field effect (left) and using our approach focusing on near structures (center) and far away structures (right).

ABSTRACT

In this paper, we introduce coverage-based opacity estimation to achieve Depth of Field (DoF) effects when visualizing molecular dynamics (MD) data. The proposed algorithm is a novel object-based approach which eliminates many of the shortcomings of state-of-the-art image-based DoF algorithms. Based on observations derived from a physically-correct reference renderer, coverage-based opacity estimation exploits semi-transparency to simulate the blur inherent to DoF effects. It achieves high quality DoF effects, by augmenting each atom with a semi-transparent shell, which has a radius proportional to the distance from the focal plane of the camera. Thus, each shell represents an additional coverage area whose opacity varies radially, based on our observations derived from the results of multi-sampling DoF algorithms. By using the proposed technique, it becomes possible to generate high quality visual results, comparable to those achieved through ground-truth multi-sampling algorithms. At the same time, we obtain a significant speedup which is essential for visualizing MD data as it enables interactive rendering. In this paper, we derive the underlying theory, introduce coverage-based opacity estimation and demonstrate how it can be applied to real world MD data in order to achieve DoF effects. We further analyze the achieved results with respect to performance as well as quality and show that they are comparable to images generated with modern distributed ray tracing engines.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

*email: {sathish.kottravel|martin.falk|erik.sunden}@liu.se

†e-mail: timo.ropinski@uni-ulm.de

1 INTRODUCTION

To better understand the underlying mechanisms of life, it is essential to investigate the structure of molecules. Considering the structure-follows-function paradigm, the understanding of a protein’s structure for instance can give essential hints about its role in metabolic pathways. Furthermore, when dealing with more than one molecule, their structure can give us clues about potential binding sites for molecule interactions. Besides those cases requiring depictions of individual molecules, visualizing the multitude of molecules within the crowded environment of the cell also calls for improved spatial comprehension. While modern life science technologies result in detailed molecular models as well as simulations containing hundreds of thousands of instances, enabling an improved spatial comprehension of these models can be challenging. As a consequence, modern visualization algorithms often struggle with the complexity of the data, which renders the application of advanced techniques tailored for improved spatial comprehension difficult.

In recent years, several visualization algorithms have been developed for improving the spatial comprehension of complex data sets, as they are acquired through imaging or result from simulations. These algorithms can be roughly classified into two groups: physically-inspired and illustrative algorithms. While physically-inspired algorithms are often based on the fundamental physical principles underlying light transport, illustrative algorithms borrow from traditional hand drawn depictions. In this paper, we tackle the problems related to spatial comprehension of large molecular data sets, by proposing a physically-inspired DoF algorithm which has specifically been developed for molecular data sets as illustrated in Figure 1. DoF techniques emerged from computer graphics, where they are often used to generate more realistic renderings. By depicting objects sharp in the focus area and increasing their blur depending on the distance to the focus area, these algorithms mimic the behavior of real world camera lenses.

As DoF is so widely used in photography, it has also received attention within scientific visualization for Focus+Context (F&C) [8, 23, 22] purposes. F&C is a general concept for directing the

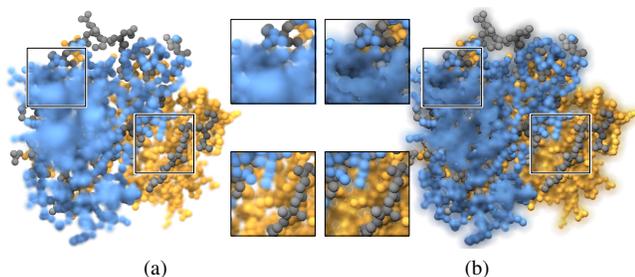


Figure 2: Comparison of our method with an image-based DoF approach. (a) our coverage-based DoF. (b) image-based DoF by post-processing depth and color buffer. Even for such relatively small molecular structures the image-based approach suffer from severe artifacts that are not visible in our object-based approach.

viewers attention onto certain parts of picture, and in scientific visualization its usage is motivated by exploration of large complex structures. Furthermore, as large complex structures normally contain small and large varying spatial locations, an adaptive F&C based on depth such as DoF becomes essential, as the focus needs to be controlled to correctly visualize the data in the desired context. More recently, DoF techniques have also been utilized in visualization where it has been shown that they have a positive effect on the depth perception of complex scenes [4].

While volumetric DoF techniques can be directly integrated into the volume rendering process [20], image-based post-processing techniques are often used for surface-based models. This is due to the fact that the physically-correct multi-sampling approach, which directly takes into account the lens geometry, often forbids interactive frame rates due to the increased sampling complexity, while the image-based post-processing effects can be applied in real-time. Unfortunately, they introduce artifacts, such as incorrect DoF effects along the silhouettes of occluding objects (see Figure 2). As inter-atom occlusions are predominant in molecular data sets, it is mandatory to resolve this issue while still allowing for interactive data exploration. To avoid this artifacts and still enable interactive frame rates, we propose a novel object-based algorithm for realizing DoF effects for MD data sets.

To address the shortcomings related to image-based DoF effects, we analyzed the nature of DoF effects as generated by a physically-correct multi-sampling renderer, and derived an object-based algorithm from these observations. The core idea of the proposed algorithm is to exploit the observed analogy between semi-transparency and the blur inherent to DoF effects. Therefore, we introduce a reformulated circle of confusion (CoC) calculation, which can be separated into two components. The first component is solely dependent on the optical parameters of the camera, while the second component only depends on an object’s location with respect to the focal distance. By exploiting this calculation, we can achieve object-based DoF effects with a single sample per pixel only. While this approach results in significantly faster rendering times as compared to multi-sampling approaches, we will show that the visual results match those which can be achieved with modern, physically-correct DoF renderers.

2 RELATED WORK

DoF rendering techniques were pioneered by Potmesil and Chakravarty with their discussion of approximating the thin lens model [17]. They developed a complex camera model for producing synthetic images which serves as an alternative for simple pin-hole camera projections. Later, Cook et al. applied distributed ray tracing to produce DoF effects by randomly sampling multiple rays

per pixel on the surface of a lens [2]. Thus, they could generate accurate DoF effects by simulating light transport between the camera and the scene whereby the pixel contributions are calculated across multiple samples instead of one ray originating from a single point. As multiple rays result in varying visibility for each pixel, this method correctly considers partial occlusion between objects. In particular, this method inherently handles some of the common artifacts that can occur during DoF computation such as intensity leakages and depth discontinuities which occur in modern image-based methods [3]. However, in order to obtain accurate results with multi-sampling, a high number of samples is required, which renders this method computationally too expensive for real-time applications. To tackle these performance problems, an alternative approach was formulated by Haeberli and Akeley to approximate DoF effects by using the z-buffer together with an accumulation buffer [7]. Instead of casting multiple rays, a scene is rendered in several passes from different positions on the lens and the accumulated result is blended. While a high number of passes leads to accurate DoF results, this method can generate convincing DoF effects at interactive frame rates by using fewer passes.

In subsequent years, several methods were developed to approximate DoF computations with the goal to obtain real-time rendering. This new stream of DoF rendering algorithms is referred to as image-based, post-filtering methods. This group of techniques exploits blurring to obtain DoF effects, whereby the blur can be achieved by using a single or multiple layers. In single layer approaches a scattering technique can be used in which the pixel values are scattered to neighboring pixels within the CoC in image space. Then the scattered pixels are accumulated and blended after depth sorting. Shinya used random multi-sampling to achieve jittering that circumvents the partial occlusion problem which demands additional computational costs [21]. As an alternative to the scattering approach, blurring can also be achieved by gathering pixel values from neighboring pixels within the CoC. As gathering is realized by using filtering techniques, visual artifacts like intensity leakages are almost unavoidable. Both, scattering and gathering as single layer, post-filtering methods, do not address blurring discontinuities. With the aid of multi-layer approaches it is possible to address partial occlusion artifacts effectively by taking the object space into account. Therefore, the scene is decomposed into several distinct layers that do not overlap in depth. Each of these layers is then blurred before the layers are composited in back-to-front order. Based on this approach many methods have been proposed to achieve DoF effects, e.g., [14, 18]. Instead, our method is related to the scattering paradigm. However, we avoid explicit scattering of pixels to neighbors by associating object silhouettes with the CoC and by using our coverage-based opacity estimation in the silhouette area. Thus, the proposed algorithm works in object space and allows partial occlusions to affect the rendering result, as it happens in ground truth multi-sampling DoF solutions.

High quality renderings of opaque complex molecular structures has been achieved in real-time for quite some time [15]. However, representations of DoF effects have been largely omitted. To our knowledge, the only exception is the work by Kauker et al., which introduces a so-called *Puxel* data structure as a discretized scene representation [12]. The authors briefly mention that by combining this data structure with quads of different sizes, a blurring which simulates DoF effects could be achieved. However, in contrast to our technique, they do not address how to capture the desired multi-sampling effects.

3 THEORETICAL BACKGROUND

DoF can be defined as a region in object space where the object of interest within the DoF appears sharp, while objects that are not in the DoF appear blurry when transformed to image space. The degree of blurriness varies with the distance from the DoF, hence it

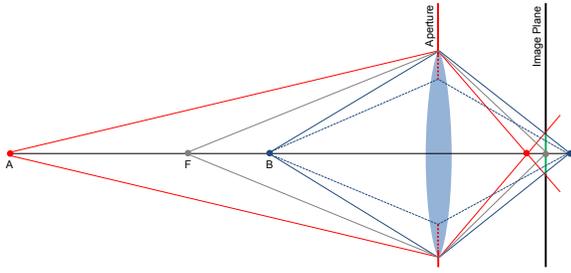


Figure 3: An illustration of the thin lens model. While point F being in focus projects to a point, points A and B that are not in focus are captured as blurred circle in the image plane. This circle is known as circle of confusion (CoC) (green). A smaller aperture results in a smaller CoC (dotted blue lines).

is possible to visually separate foreground and background in captured images. In photography, DoF effects are an important aspect in directing the focus of viewers to a specific region in an image and known for modifying various characteristics such as the aesthetic quality of an image. Furthermore, DoF effects are interpreted as depth cues by the human visual system, which makes DoF an effective way to improve spatial comprehension [9]. In the past, extensive studies have been conducted on various depth cues and on DoF in particular [14, 4].

3.1 Thin Lens Model

Although, photography cameras are based on complex optical systems, in computer graphics often a thin lens model is used to produce photo-realistic images [17]. Part of the simplicity of this model results from the ray optics representations using a Gaussian lens formula

$$\frac{1}{f} = \frac{1}{d} + \frac{1}{d'}, \quad (1)$$

where f is the focal length of the lens, d is the distance to the object in front of the lens, and d' is the distance to the image sensor or image plane behind the lens as shown in Figure 3. This model enables us to control the DoF using various parameters such as aperture size, focal distance and focal length of a lens, distance to the image sensor, as well as size of the image sensor and distance to the object. In Figure 3, point F is at focal distance d_{focus} and the image plane lies at d'_{focus} . Accordingly, by using (1) the distance to the image plane d'_{focus} can be computed as follows:

$$d'_{\text{focus}} = \frac{fd_{\text{focus}}}{d_{\text{focus}} - f}. \quad (2)$$

The points A and B are out of focus, and thus appear as blurred circles, the CoC, on the image plane. Their images would appear sharp in front and behind of the image plane, respectively. The diameter of the CoC depends on the distance of an object in the lens as well as the aperture diameter. The aperture d_a diameter is expressed as f-number $a = \frac{f}{d_a}$. The change in diameter of the CoC d_{coc} for any point at distance d in front of the lens can be computed [17] according to the following equation:

$$d_{\text{coc}}(d) = \left| \frac{fd}{d-f} - \frac{fd_{\text{focus}}}{d_{\text{focus}}-f} \right| \left(\frac{d-f}{ad} \right). \quad (3)$$

Based on this equation it can be seen that the CoC for a given distance d is only dependent on the focal length, the aperture, and, of course, the focus distance. Figure 4 shows the diameter of the CoC for a given parameter set. At the focal distance d_{focus} of 1m, the CoC function is discontinuous and evaluates to zero, i.e. yielding a

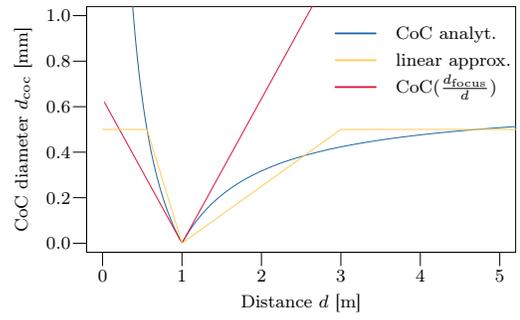


Figure 4: Diameter of the circle of confusion (CoC). When the CoC is calculated for the ratio of d_{focus} to d , the CoC is proportional to this ratio (red). The piecewise linear approximation of the CoC (orange) is used in the approach by Scheuermann and Tatarchuck [19]. Parameters: $f = 35$ mm, $a = f/2$, and $d_{\text{focus}} = 1$ m.

sharp image on the image plane. For smaller values, the diameter increases drastically whereas for larger distances the CoC diameter converges. To prevent the evaluation of Equation 3, the CoC diameter is often approximated with a piecewise linear function defined through four segments [19].

4 OUR APPROACH

Based on the theoretical background discussed above, we were able to come up with an alternative DoF formulation, which has specifically been developed for molecular visualization. In this section, we detail this formulation by first describing the enabling observations which we could derive from a physically correct DoF rendering process.

4.1 CoC Calculation

In contrast to previous work, we propose to use the precise CoC diameter while at the same time reducing the computational cost for evaluating Equation 3. This can be achieved by exploiting the following observation. When considering the CoC diameter of the ratio d_{focus} to d instead of simply using the distance d , a linear relation emerges (see red line in Figure 4). More interestingly, the slopes of both lines are equal, apart from the sign. Using (3) and let d be $\frac{d_{\text{focus}}}{2}$, the slope m is given by:

$$m = \left| \frac{f}{1 - 2\frac{f}{d_{\text{focus}}}} - \frac{fd_{\text{focus}}}{d_{\text{focus}} - f} \right| \left(\frac{1 - 2\frac{f}{d_{\text{focus}}}}{a} \right), \quad (4)$$

where the slope is independent of the distance d . To obtain the correct CoC diameter from the slope m we only have to multiply it by $\frac{d_{\text{focus}}}{d}$ and shift it along the x axis by 1, thus yielding

$$d_{\text{coc}}(d) = \left| \frac{d_{\text{focus}}}{d} - 1 \right| m. \quad (5)$$

As a consequence, the CoC calculation is reduced to a mere subtraction, one division, and one multiplication with the precalculated slope. Please note that the result of (5) is identical to (3) and thus matches the blue curve in Figure 4. This simplification is generally applicable not only to our approach but also to other DoF schemes like the one proposed by Scheuermann and Tatarchuck [19].

4.2 DoF Multi-Sampling

Rendering engines relying on ray tracing typically employ a more or less sophisticated lens model to include DoF effects, e.g., the thin lens model described above. As described in Section 2, most of these renderers use a gathering approach, whereby multiple rays

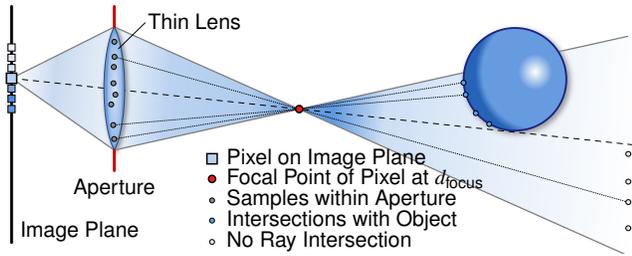


Figure 5: Multi-sampling DoF as often computed in ray tracing. The DoF effect is obtained by sampling rays starting inside the open part of the aperture of the thin lens. All rays are directed toward the point in the scene which would produce a sharp image in the pixel (red dot) and intersection tests are performed.



Figure 6: Reference images for Phong illumination generated by the Mitsuba renderer. Sphere without (left) and with DoF effect (right).

per pixel are shot into the scene collecting color information which is then merged. Figure 5 illustrates this multi-sampling approach for a single pixel. The lens area within the open aperture is covered by random samples which serve as origins for the rays. Depending on the number of samples the resulting image will feature some noise, whereby the more samples are used the less variation is contained in the result. The focal point lying on the view ray originating from the pixel, i.e., the position at the focal distance d_{focus} , is used to determine the direction of the sampled rays. Thus, all rays are restricted to a double cone whose width corresponds to the CoC. Intersection tests are carried out for each ray and the final color of the pixel is determined by averaging the results. In the example of Figure 5, four samples intersect with the spherical object whereas the remaining four samples hit nothing. This results in a mixture of object color and background to equal parts. We will elaborate on this effect and exploit it in the next section.

4.3 Coverage-Based Opacity Estimation

If a photograph is taken, objects outside the DoF region appear blurry and transparent, which can also be observed in artificial scenes rendered with DoF. As seen in Figure 6, the outlying regions of a sphere appear as if they are rendered semi-transparently since the parts of the background are also contributing to the pixel colors. As described above, employing multi-sampling for DoF averages the individual ray contributions with equal weights, whereby a coverage of half the samples leads to 50% of the contribution (cf. Figure 5). If it is possible to confirm a correlation of the coverage and the apparent transparency, we could estimate at any point the opacity of a certain object with respect to its CoC.

To investigate this correlation, we set up a scene containing a white disk facing the camera and a black background. The disk is positioned so that the view is centered on its fringe as depicted in Figure 7 (a). The scene was rendered with the physically-based Mitsuba renderer [10] at a resolution of 512×512 pixels using 1024 samples per pixel whereby only the lens aperture has been changed in this experiment (Figure 7 (b)-(d)). Figure 7 (bottom) shows a plot

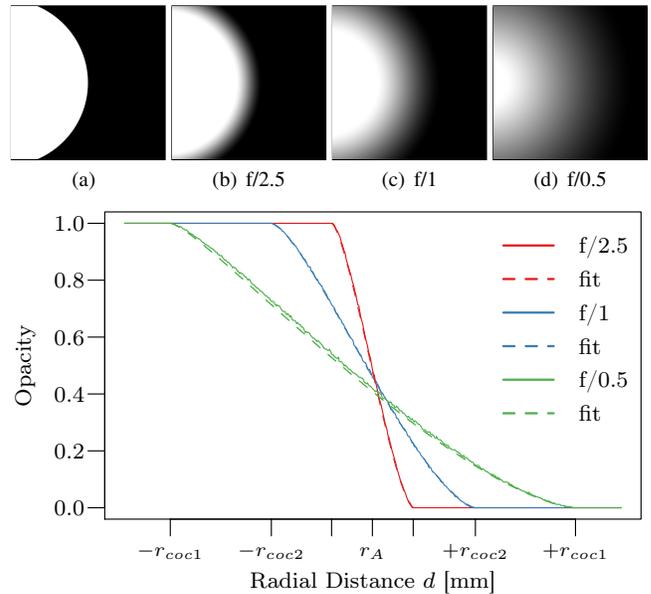


Figure 7: Reference DoF images generated by the Mitsuba renderer. (a) pinhole camera. (b) small aperture, $f/2.5$, $d_{\text{coc}} = 5.45$ mm. (c) medium aperture, $f/1$, $d_{\text{coc}} = 15.79$ mm. (d) large aperture, $f/0.5$, $d_{\text{coc}} = 42.86$ mm. The radial opacity values measured in (b)-(d) are plotted together with our approximation (bottom). r_A and r_{CoC} are atom radius and CoC disk radius respectively. Parameters: $f = 50$ mm, $d_{\text{focus}} = 0.12$ m, object distance $d = 0.3$ m, and disk diameter $r = 0.1$ m.

of the pixel intensities of a central horizontal lines running through each of the resulting images. Since the disk features a constant color, we can directly interpret the data values as opacity. The slight variation in the signals is caused by randomized sampling.

Following the notion of area coverage, we can describe the radial opacity of the disk with the help of a sampling point moving radially outward. The CoC disk is centered at the sampling point and the CoC radius is determined from the object's distance to the camera. As long as the CoC disk is entirely covered by the object, the result is fully opaque. Similarly, as soon as the CoC disk no longer intersects with the object, the object no longer contributes and, thus, appears transparent for these sample positions. In between, the opacity varies as does the overlap between both, object and CoC disk. Since the object is a circular disk, we can compute the overlap A as the sum of two circle segments. By normalizing the overlap A with respect to the area of the CoC disk we obtain a local coverage measure. When we apply this measure to our test scene, this analytical solution matches nicely with the results obtained from the renderings (see Figure 7, bottom). Hence, we can conclude that it is possible to relate the blur of DoF effects to a coverage-based transparency representation. Please note that the maximum opacity is determined by the ratio of object area to CoC disk area.

The concept of using a disk-disk intersection to compute the opacity value for a DoF calculation can also be applied to spherical objects. Please note that the overlap calculation does not take the spatial extent along the viewing direction into consideration. Figure 8 shows the difference in our experimental scene where the difference between a disk and a sphere with the same radius was about 10% in the outer regions. However, in our main application, the visualization of molecular data, this error is hardly noticeable due to the small atom radii in combination with blending.

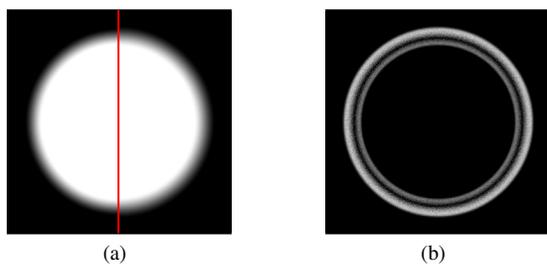


Figure 8: Comparison of DoF applied to a disk and a sphere. (a) disk (left) and sphere (right). (b) difference between disk and sphere (scaled by a factor of 10 for illustration purposes). Images have been created with the Mitsuba renderer.

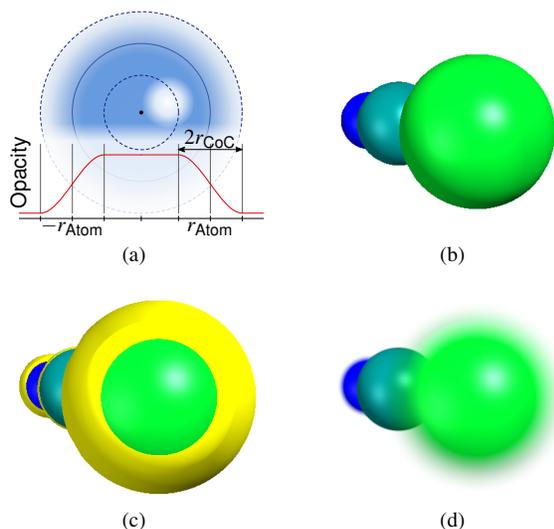


Figure 9: Illustration of the opacity fall-off function. (a) the fall-off function is applied to the enlarged glyph footprint. (b) atoms rendered without DoF effect. (c) enlarged glyphs with area affected by fall-off indicated in yellow. (d) final rendering with alpha blending yielding the DoF effect.

5 DOF IN MOLECULAR VISUALIZATION

In principal the derived coverage-based DoF representation is applicable to arbitrary geometry. However, within the context of this paper we apply it to molecular visualizations, more specifically to glyph-based atom rendering. Therefore, we realize our coverage-based DoF representation by increasing the radius r_{Atom} of an atom by the CoC radius, r_{CoC} . This increment is neglected for atoms lying in the focal plane since their computed r_{CoC} radius is close to zero. The enlarging of the glyph silhouette is done at each atom position which is the center of its silhouette. This results in a silhouette extent region in addition to the original silhouette of the atom, i.e., $r_{ext} = r_{Atom} + r_{CoC}$. For the region within $r_{Atom} - r_{CoC}$ and $r_{Atom} + r_{CoC}$ as shown in Figure 9, we then introduce an opacity fall-off modeled according to our observations shown in Figure 7. We call the region covering the fall-off in image space the *blur disk*. Due to its semi-transparent nature, the fall-off opacity will permit atoms lying behind the blur disk to be visible (cf. Figure 9). Consequently, it enables to see through semi-transparent points in unfocused regions, which resembles the partial occlusion effects previously only achieved through multi-sampling DoF methods (cf. Section 4.2). Figure 10 illustrates this see through based on a simple

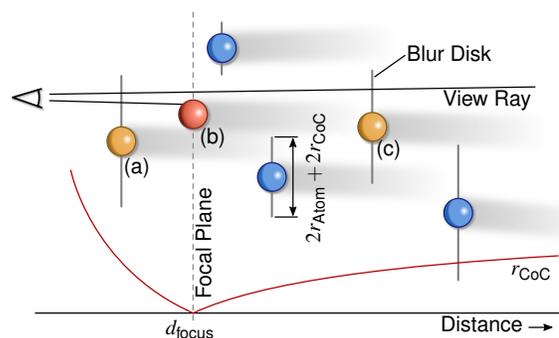


Figure 10: The glyph footprint is radially enlarged by twice the CoC radius r_{CoC} depending on the distance to the camera and lens parameters. If a view ray hits a blur disk, the respective glyph contribution is calculated from a single sample. Glyphs in front of the focal plane are spread out over the atom boundaries (a). Objects in focus (b), i.e. $d = d_{focus}$, do not possess a blur disk and are treated as entirely opaque. Although glyphs might be occluded (c), they still can contribute to view rays due to their enlarged blur disk.

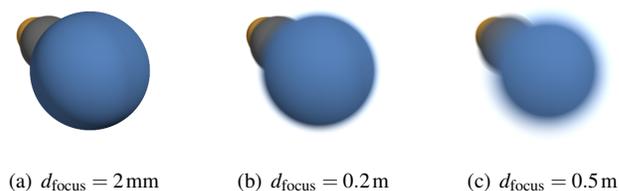


Figure 11: The contribution of occluded glyphs is considered by employing the opacity fall-off function. The focal distance increases from front to back (a) to (c). With larger focal distance and, thus, increasing CoC diameter the occluded atoms become visible (Parameters: $f = 0.28$ mm, $a = f/0.1$).

scene that has atoms with the CoC silhouette extent varying with respect to the focal plane. A ray cast through the scene can pass through the blur disk and gather contributions along the ray for the respective screen pixel.

Figure 11 shows the results of applying our approach to a simple scene containing three atoms. The scene consists of a blue atom in the front occluding a gray atom in the middle and an orange one in the back. As it can be seen our coverage-based opacity estimation realizes DoF effects in a convincing way, as it also captures partial occlusions. We would like to point out again that this contribution of occluded structures is not possible with traditional image-based methods, while our method can capture these effects and thus enables to achieve an image quality similar to those of multi-sampling methods.

6 IMPLEMENTATION

Our approach is implemented in C++ and OpenGL with GLSL. The glyphs are rendered using a billboard approach conducted in screen space where the silhouette of an atom is determined [6, 5], whereby within each silhouettes we perform ray-sphere intersections. This technique enables us to perform ray casting for each silhouette in the fragment shader. During this process, we compute depth, normal, and intersection points within the silhouette region, which are used for illumination of the three dimensional spheres representing atoms [5]. Each atom glyph is aware of its extended radius r_{ext} and also the radius r_{Atom} of the original atom.

Using the extended radius r_{ext} , we can construct a spherical sur-

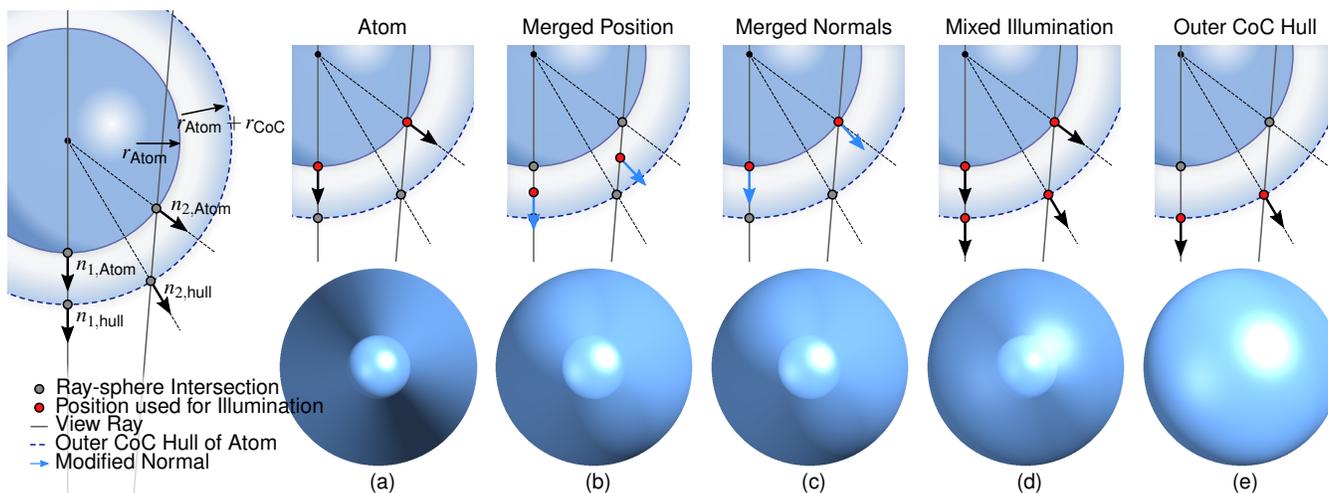


Figure 12: Comparison of different approaches for atom shading. (a) regular Phong shading on the inner sphere. Normals extending outward from the inner sphere are used to shade the outer sphere. (b) ray intersections with the inner and the outer sphere are interpolated before shading is computed. (c) normals of the inner sphere and the outer sphere intersection are merged before shading. (d) shading computation is performed for both, inner sphere and outer sphere intersection, before being blended. (e) only intersections with the outer sphere are used.

face which forms the CoC hull of the blur disk and with r_{Atom} we can construct the true spherical surface of the atom. We refer to the original atom with r_{Atom} as the inner sphere, whereas the extended CoC hull is denoted as outer sphere.

Atom rendering. As our DoF representation exploits semi-transparency, depth sorting is required to obtain a correct blending. In our initial implementation, we have used per-pixel linked lists [24] (PPLL) to achieve depth sorting for the fragments generated for all inner and outer spheres. Unfortunately, for complex scenes this lead to an overflow of our lists, which resulted in visual artifacts. Therefore, we have reduced the entries to be written to the linked lists to those fragments generated by the outer spheres only. As these are the only semi-transparent structures in the scene, we can initially generate a depth buffer image for the inner spheres, and discard the fragments of the outer spheres in a z-buffer test against this depth buffer. This procedure reduces the number of entries in the linked lists drastically, and thus enables artifact-free rendering even for complex scenes.

Thus, our algorithm initially renders the inner spheres into a depth and a color buffer. Based on the resulting depth buffer, the outer spheres are processed, whereby those fragments lying in front of the closest inner sphere fragment are written to a pixel linked list. In the next pass, these PPLL are sorted, and the semi-transparent fragments are used as entry and exit points in a fragment shader to compute the semi-transparent contributions lying in between them. Thus, the sorting results in correct blending of the semi-transparent structures and thus achieves convincing DoF effects. While the depth values in the first pass can be computed in the same manner as in the original glyph-based ray-casting, computation of shading values requires some special considerations.

Shading computation. When computing the shading for inner and outer spheres, it is essential that no shading discontinuities appear. Therefore, an appropriate shading transition is required between these spheres. Figure 12 (a) and (e) show shading discontinuities as they appear when only using the normals of the inner sphere (a) or the outer sphere (e) for the shading computation. To avoid these discontinuities, we propose different normal interpolation schemes which take into account the normals and intersection points at the inner sphere (atom) and the outer sphere (hull):

- **Merged position.** Compute the intersection points on both outer and inner spheres. Interpolate a new intersection point p

between the computed intersection points. Obtain the normal at p and compute the shading (see Figure 12 (b)).

- **Merged normals.** Compute intersection points on both outer and inner spheres. Obtain individual normals at each intersection. Interpolate new normal from the obtained normals and compute the shading (see Figure 12 (c)).
- **Mixed shading.** Compute the intersection points of the outer and inner sphere. Obtain individual normals at each intersection. Compute the shading at each intersection point separately and blend the results (see Figure 12 (d)).

As Figure 12 (d) delivered the most convincing results in all cases, we have used mixed shading in our implementation.

7 RESULTS & DISCUSSION

Image quality. As shown in Figure 10 in our approach view rays pass through the expanded glyphs and hit other glyphs lying behind the blur disk. As a consequence, the coverage-based opacity fall-off applied to the outer regions of the enlarged atoms simulates a realistic blur effect, which is usually only achievable by multi-sampling based DoF renderers. To illustrate the applicability of our method to real world MD data sets, we use data provided by the RCSB protein data base (PDB) [1, 16]. The used colors in all our results are assigned based on the residue color schemes used in standard molecular visualization packages, such as Jmol [11]. Figure 13 compares the result of our interactive approach with the outcome of using a physically-based ray tracing in the Mitsuba renderer. We used 2,500 samples per pixel in Mitsuba to suppress most of the sampling-related noise requiring a total render time of about 12 minutes. Reducing the number of samples to 625 per pixel reduces the render time to 3 minutes while featuring some noise. Figure 15 depicts different DoF settings within our algorithm for the human protein kinase (MEK2) and a ribosome (Thermus thermophilus 70S). Another molecular structure of the ribosome is depicted in Figure 1. **Rendering performance.** Table 1 shows the frame rate measured on an NVIDIA GeForce GTX TITAN. The molecular data is rendered with three setups in order to measure performance. First setup is PPLL disabled and without DoF. Second setup with PPLL enabled and without DoF. Finally the same data is rendered PPLL enabled and with DoF. The overhead for initializing the PPLLs affected our performance significantly, while the computation of the

Table 1: Performance measurements represented in the table are taken by rotating the camera around the scene. The rotation is performed in 60 steps around two orthogonal axes where each axis is changed in steps by $\frac{2\pi}{30}$, respectively.

PDB Dataset (ID)	# Atoms	FPS (512 × 512)			FPS (1024 × 1024)			
		No-PPLL + No-DoF	PPLL + No-DoF	PPLL + DoF	No-PPLL + No-DoF	PPLL + No-DoF	PPLL + DoF	
3UF1	3527	80.9	25.1	21.2	22.1	14.4	11.1	
1S9I	4686	79.6	24.3	21.52	20.62	7.1	5.01	
2WDG	56532	63.88	21.35	14.58	29.09	6.83	4.69	
2WDK	147236	62.3	17.68	13.2	25.58	5.48	4.28	

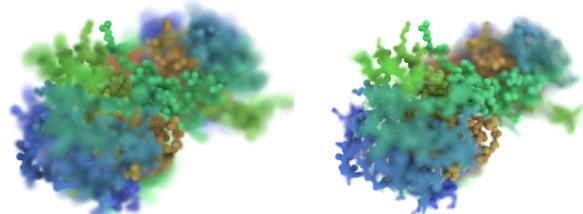


Figure 13: Comparison of our approach (left) with a physically based rendering in Mitsuba (right), human protein kinase MEK2. (PDB ID: 1S9I). At 2,500 samples per pixel Mitsuba took approximately 12 minutes to render. Our approach took 1.5 ms.

DoF effect itself consumed rather little time. As discussed in Section 6, exploiting the z-buffer test to identify the PPLL objects has shown significant reduction in GPU memory usage (see Figure 14).

Application and expert opinion. We applied our DoF effect in three exploration scenarios. In the first scenario, free 3D navigation was used together with a fixed focus distance. In the second scenario, the user could point and click to define the focus distance with respect to the selected structure. In the third scenario, focus distance is varied sequentially to follow or focus a series of atoms in a static position. We have applied this mode as a semantic DoF effect [13] to charge transport simulation data, where the goal is to follow the flow of charge through molecule residues of solar panel cells.

We questioned domain experts regarding the usefulness and applicability of DoF in this context. They stated that they like the idea of focusing on a small region within one molecule, while being also able to apply a semantic DoF for highlighting individual atoms. Another expert remarked that in his opinion DoF is of particular importance for attention guidance when trying to show or explain certain features to another person.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced coverage-based opacity estimation for interactive DoF in molecular visualization. Based on observations made from physically-correct DoF effects, we were able to propose a novel DoF algorithm which exploits object semi-transparency instead of object blur. The algorithm is specifically tailored towards molecular visualizations, where it exploits the instance-based nature of the data to be visualized. By only requiring a single sample per pixel our algorithm supports interactive visualization, while not suffering from the occlusion-based artifacts often encountered when dealing with image-based DoF algorithms (cf. Figure 2). We have applied the algorithm to several real-world data sets and could show that the image fidelity is comparable to physically-based solutions.

While the proposed algorithm enables us to achieve convincing results, we see several opportunities for future work. Our method

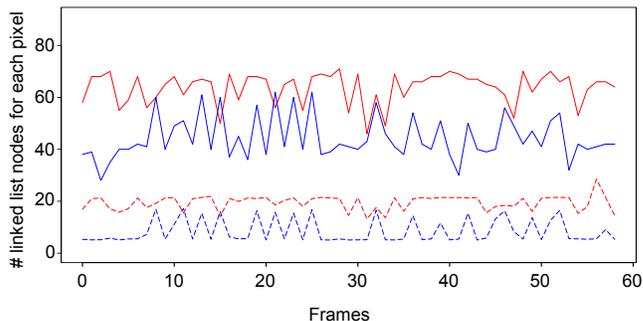


Figure 14: Per-pixel linked-list node count captured for 60 frames with DoF enabled. The maximum linked-list count is shown for each frame without (red) and with depth test (blue) against the inner atom (lower values are better). Mean values are depicted in dashed lines.

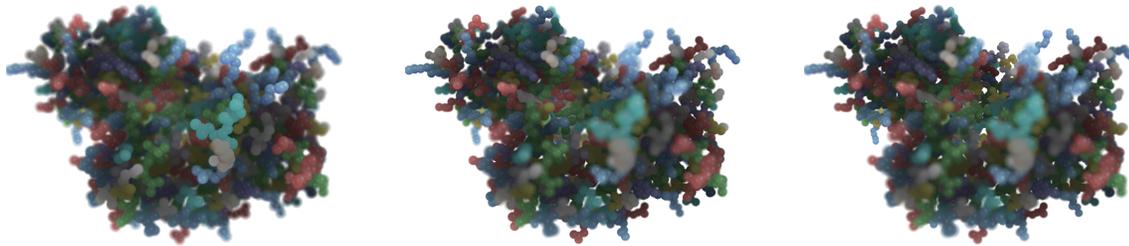
can be easily adapted to molecular representations such as stick and space-filling models, but it could be challenging to apply it to ribbon models which involve complex geometry. While van-der-Waals spheres are one of the simpler molecule representations, we would like to investigate the application to more complex representations, such as solvent-accessible surface or solvent-excluded surfaces. As both of these representations can be achieved by using the rolling-ball metaphor, we are optimistic that the presented concepts can be helpful. Besides this, we would also like to investigate in which areas general geometries can benefit from our approach, when it is applied in areas outside molecular visualization. In this case, the challenge is to realize the outer shell of individual structures. One possibility is to deduce a distance map from individual structures rendered as image masks and use the contour of the deduced distance map to obtain the outer shell. At the moment our illumination compensation approach is based on a linear interpolation scheme. To be able to deal with textured surfaces, we would like to investigate how more complex mechanisms can be exploited to deal with richer surface details while at the same time improving the quality for specular highlights.

9 ACKNOWLEDGMENTS

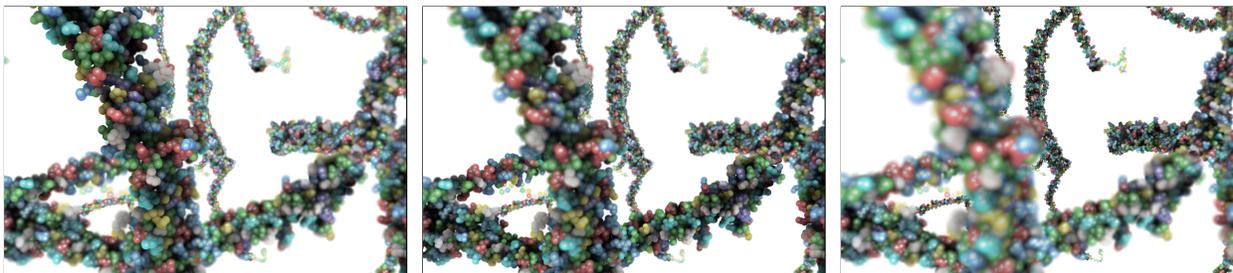
This work was supported through grants from the Excellence Center at Linköping and Lund in Information Technology (ELLIIT) and the Swedish e-Science Research Centre (SeRC). The presented concepts have been developed and evaluated in the Inviwo framework (www.inviwo.org).

REFERENCES

- [1] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.
- [2] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (Proceedings of SIGGRAPH '84)*, 18(3):137–145, 1984.



(a) Human protein kinase MEK2 (PDB ID: 1S9I), $f = 5.98$ nm, $a = 0.61$ nm, $d_{\text{focus}} = 5.51$ nm (left), $d_{\text{focus}} = 6.37$ nm (middle), $d_{\text{focus}} = 7.93$ nm (right).



(b) Multiple instances of one half of the rat liver vault protein (39mer of the major vault protein, PDB ID: 2ZUO, 2ZV4, 2ZV5), $f = 2$ nm, $a = f/0.3$, $d_{\text{focus}} = 16$ nm (left), $d_{\text{focus}} = 30$ nm (middle), $d_{\text{focus}} = 94$ nm (right).

Figure 15: Depth of field applied to different protein structures. (a) Human protein kinase MEK2 (PDB ID: 1S9I). (b) Half vault protein of the rat liver (PDB ID: 2ZUO, 2ZV4, 2ZV5).

- [3] J. Demers. Depth of field in the toys demo. In *Game Developers Conference*, 2003.
- [4] A. Grosset, M. Schott, G.-P. Bonneau, and C. D. Hansen. Evaluation of depth of field for depth perception in DVR. In *IEEE Pacific Visualization Symposium*, pages 81–88, 2013.
- [5] S. Grottel, G. Reina, and T. Ertl. Optimized data transfer for time-dependent, GPU-based glyphs. In *IEEE Pacific Visualization Symposium*, pages 65–72, 2009.
- [6] S. Gumhold. Splatting illuminated ellipsoids with depth correction. In *International Fall Workshop on Vision, Modelling and Visualization*, pages 245–252, 2003.
- [7] P. Haerberli and K. Akeley. The accumulation buffer: hardware support for high-quality rendering. *Computer Graphics (Proceedings of SIGGRAPH '90)*, 24(4):309–318, 1990.
- [8] H. Hauser. Generalizing focus+context visualization. In *Scientific Visualization: The Visual Extraction of Knowledge from Data*, Mathematics and Visualization, pages 305–327. Springer Berlin Heidelberg, 2006.
- [9] I. P. Howard. *Seeing in Depth, Vol. 1: Basic Mechanics*. University of Toronto Press, 2002.
- [10] W. Jakob. Mitsuba—physically based renderer. <http://www.mitsuba-renderer.org/>.
- [11] Jmol: an open-source java viewer for chemical structures in 3D. <http://www.jmol.org/>.
- [12] D. Kauker, M. Krone, A. Panagiotidis, G. Reina, and T. Ertl. Rendering molecular surfaces using order-independent transparency. In *Proceedings of the 13th Eurographics Symposium on Parallel Graphics and Visualization*, EGPGV '13, pages 33–40, 2013.
- [13] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *IEEE Symposium on Information Visualization (INFOVIS 2001)*, pages 97–104, 2001.
- [14] S. Lee, G. Jounghyun Kim, and S. Choi. Real-time depth-of-field rendering using anisotropically filtered mipmap interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):453–464, 2009.
- [15] L. Marsalek, A. Dehof, I. Georgiev, H.-P. Lenhof, P. Slusallek, and A. Hildebrandt. Real-time ray tracing of complex molecular scenes. In *Information Visualization (IV), 2010 14th International Conference*, pages 239–245, July 2010.
- [16] Rcsb protein data bank. <http://www.pdb.org/>.
- [17] M. Potmesil and I. Chakravarty. A lens and aperture camera model for synthetic image generation. *Computer Graphics (Proceedings of SIGGRAPH 1981)*, 15(3):297–305, 1981.
- [18] D. C. Schedl and M. Wimmer. A layered depth-of-field method for solving partial occlusion. *WSCG*, 20(3):239–246, 2012.
- [19] T. Scheuermann and N. Tatarchuk. Improved depth of field rendering. In W. Engel, editor, *ShaderX³*, chapter 4.4, pages 363–377. Charles River Media, 2004.
- [20] M. Schott, A. Pascal Grosset, T. Martin, V. Pegoraro, S. T. Smith, and C. D. Hansen. Depth of field effects for interactive direct volume rendering. *Computer Graphics Forum*, 30(3):941–950, 2011.
- [21] M. Shinya. Post-filtering for depth of field simulation with ray distribution buffer. In *Graphics Interface*, pages 59–59. Canadian Information Processing Society, 1994.
- [22] I. Viola and E. Grller. On the role of topology in focus+context visualization. In H. Hauser, H. Hagen, and H. Theisel, editors, *Topology-based Methods in Visualization*, Mathematics and Visualization, pages 171–181. Springer Berlin Heidelberg, 2007.
- [23] I. Viola and M. E. Gröller. Focus+context visualization of features and topological structures. Technical report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, 2005.
- [24] J. C. Yang, J. Hensley, H. Grün, and N. Thiberoz. Real-time concurrent linked list construction on the GPU. *Computer Graphics Forum*, 29(4):1297–1304, 2010.