

A General Illumination Model for Molecular Visualization

P. Hermosilla¹, P. Vázquez², A. Vinacua², T. Ropinski¹

¹Visual Computing Group, Ulm University, Ulm, Germany
²VIRVIG Group, Universitat Politècnica de Catalunya, Barcelona, Spain

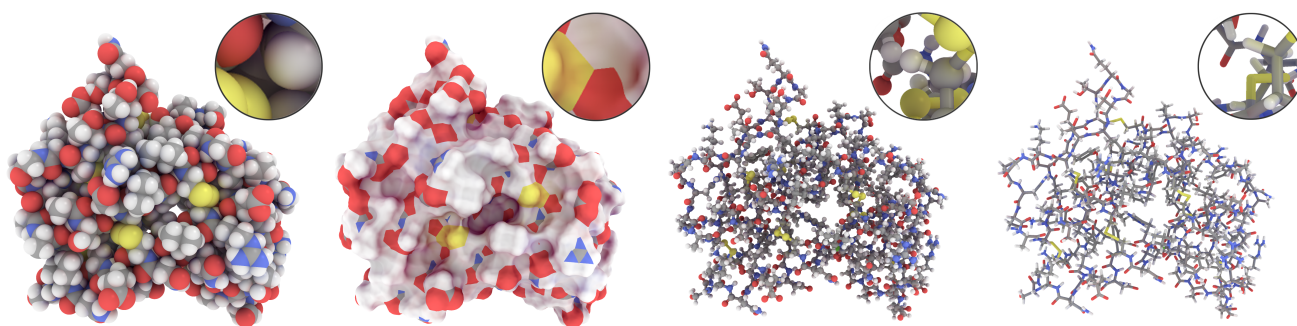


Figure 1: Example of using our new diffuse illumination model for molecules. The system uses a regression analysis over a strategically sampled set of spatial configurations to create a set of equations that can be evaluated in real time to shade molecular models based on cylinders and spheres. The images illustrate the application of our illumination model to different representations of the same molecule.

Abstract

Several visual representations have been developed over the years to visualize molecular structures, and to enable a better understanding of their underlying chemical processes. Today, the most frequently used atom-based representations are the Space-filling, the Solvent Excluded Surface, the Balls-and-Sticks, and the Licorice models. While each of these representations has its individual benefits, when applied to large-scale models spatial arrangements can be difficult to interpret when employing current visualization techniques. In the past it has been shown that global illumination techniques improve the perception of molecular visualizations; unfortunately existing approaches are tailored towards a single visual representation. We propose a general illumination model for molecular visualization that is valid for different representations. With our illumination model, it becomes possible, for the first time, to achieve consistent illumination among all atom-based molecular representations. The proposed model can be further evaluated in real-time, as it employs an analytical solution to simulate diffuse light interactions between objects. To be able to derive such a solution for the rather complicated and diverse visual representations, we propose the use of regression analysis together with adapted parameter sampling strategies as well as shape parametrization guided sampling, which are applied to the geometric building blocks of the targeted visual representations. We will discuss the proposed sampling strategies, the derived illumination model, and demonstrate its capabilities when visualizing several dynamic molecules.

CCS Concepts

•Human-centered computing → Scientific visualization; •Computing methodologies → Computer graphics; Rendering;

1. Introduction

Molecular simulations have become an important pillar of modern research in areas such as biology and chemistry, since running experiments *in silico* effectively reduces time and resources in traditional wet labs. The output of these simulations is usually one or multiple molecular trajectories composed of hundreds or thousands of steps, whereby for each step a spatial atom arrangement repre-

senting the overall shape is stored. Analyzing such data can be a tedious task without the proper tools. Visualization helps by providing insightful visual representations, highlighting specific features and allowing to form new hypothesis based on the data at hand. Nevertheless, to be beneficial it is of utmost importance that the shape of the visualized molecule is unambiguously perceivable. Unfortunately, due to the complex nature of molecules, and

the representations typically used to visualize them, unambiguous perception is not always guaranteed. To improve the situation, advanced illumination models have been proposed to increase the perception in such cases [TCM06]. While the light intensity gradient over a surface can provide enough cues to recognize the shape of an object, in cluttered images such local shading is not sufficient to understand the spatial relations of the different structures of the model. Global illumination techniques, on the contrary, are able to provide additional structural cues, and thus facilitate unambiguous perception. Color bleeding is a good example of such a technique since it provides information about the neighboring objects by simulating the reflection of light on their surfaces. Moreover, as indicated by Color Constancy studies [Fos11], it maintains a good perception of the individual objects, a crucial aspect in molecular visualizations. Unfortunately, as they simulate multiple bounces of light in a scene, global illumination models are of high computational complexity. Their recursive nature usually limits them to off-line rendering, making them not suitable for interactive molecular visualization. Nevertheless, in the past, a few techniques successfully approximated global illumination effects for molecular scenes (e.g., [SGG15, SVGR16]). However, due to the fact that existing techniques are limited to Space-filling representations only, until today it has not been possible to achieve consistent illumination when combining different visual representations or switching between them, which are tasks often performed when analyzing molecular simulation data.

In this paper, we present the first general illumination model which can be applied to the four most frequently used atom-based molecular representations, namely the Space-filling, the Solvent Excluded Surface, the Balls-and-Sticks, and the Licorice representation. Other existing representation methods, such as Ribbons, are also popular among domain experts. However, they perform a high abstraction of the molecule by removing atomic information. Whilst these models improve the analysis of certain properties of proteins, in this work, we focus on atom-based representations. The presented illumination model employs an analytical solution to the light transport problem between perfectly diffuse convex objects, which makes it applicable in real-time. To obtain our analytical solution, we analyze the high dimensional parameter space implicitly defined by the geometries of the visual representations and employ symbolic regression [SL09] to fit an adequate function. The idea of deriving an analytical solution for light transport through regression analysis is not new [RWG*13, SVGR16], but previous solutions have focused on single rendering primitives, such as triangles or spheres. While the reported outcomes for these primitives are convincing, the nature of the regression process, which involves learning from a multitude of samples, has until now forbidden the application to more complex geometries consisting of different shapes. To resolve this limitation, we propose effective sampling strategies, which drastically reduce the number of samples to be considered during regression analysis. For an effective sampling, we show first how to reduce the number of samples by limiting ourselves to a set of representative spatial configurations. Then, each configuration is reparameterized based on the nature of the shape of the function that we aim to approximate. We will show that by applying these two strategies, we arrive at a number of samples which is low enough to perform a regression analysis using symbolic regression, while still arriving at high-quality illumination models. To demonstrate the feasibility of the sampling strategies, we apply them to

obtain a general illumination model, that can be applied during an interactive visualization process. As the model describes the radiance at one point as the summation of the reflected light by the nearby objects and their visibility, it results in molecular visualizations incorporating global illumination effects. We would like to emphasize, that the proposed illumination model is only one possible area, where the improved sampling strategies enable regression analysis, and that we expect to see more such models to be derived based on these strategies. Thus, the main contributions of this paper are:

- **Sampling strategies** developed to enable regression analysis of illumination data for complex geometric structures composed out of multiple geometric primitives.
- A general **interactive illumination model** for atom-based molecular visualization, which is applicable to the Space-filling, the Solvent Excluded Surface, the Balls-and-Sticks, and the Licorice representation.

The remainder of the paper is structured as follows. Section 2 provides an overview of the work related to our approach. Section 3 formalizes light interactions between geometric primitives, such that regression analysis becomes possible. Since analyzing these interactions naively results in a prohibitive number of samples, we introduce sampling strategies to reduce the samples to a practical level in Section 4. Section 6 demonstrates how these sampling strategies can be applied to obtain a general illumination model for the targeted visual representations, i.e., Space-filling, Solvent Excluded Surface, Balls-and-Sticks, and Licorice. Then, in Section 7, the real-time algorithm that implements this model is introduced. Section 8 presents results obtained for different molecules, and Section 9 discusses the limitations of the presented approach. Finally, the paper concludes in Section 10 by summarizing the findings and discussing open future work.

2. Related Work

In this section, we will describe the most relevant methods developed to visualize the targeted molecular representations, i.e., Space-filling (SF), Solvent Excluded Surface (SES), Balls-and-Sticks (B&S), and Licorice representations, with a special focus on the perceptual enhancement through advanced illumination effects. Since providing a comprehensive review of molecular visualization techniques is beyond the scope of this paper, we would like to refer the reader to the recent state-of-the-art report published by Kozlíková et al. for this matter [KKF*16].

Molecular Representations. As different visual representations for molecules exist, they all have their individual benefits and downsides. The Space-filling model provides a good overview of the space occupied by a molecule and its cavities, by representing the atoms as spheres. However, due to this dense representation, it occludes most of the internal structure of the molecule. The Balls-and-Sticks and Licorice models, on the other hand, provide a good balance between information presented and screen foot-print. In these models, the atoms are also represented by spheres and the bonds between atoms are represented by cylinders. In these models the size of the objects is smaller than in the Space-filling, leaving thus enough empty space to visualize the surrounding objects. Solvent Excluded Surfaces are derived from the Space-filling model and, therefore, they present the same benefits. Besides those, they also provide information about the accessibility of the molecule.

While many visual representations of molecules employ spheres to depict atoms, the most commonly used technique to render these spheres is by using glyphs [Gum03, RE05, TCM06, SWBG06]. However, some authors used more sophisticated techniques in order to render millions of these particles in real time. Falk et al. [FKE13] presented a system which was able to render large-scale molecular systems by ray-casting a grid-based data structure. Le Muzic et al. [LMPSV14] also developed a technique which is able to render molecular aggregations in real-time. By exploiting the capabilities of modern GPUs, they were able to generate different levels-of-detail on-the-fly. While sphere rendering is sufficient for the SF representation, B&S and Licorice require also the integration of cylindrical primitives, which are also most commonly rendered using glyphs. Tarini et al. [TCM06] and Sigg et al. [SWBG06], both presented in 2006 two such algorithms, while the technique presented by Tarini et al. was limited to orthographic projections. The most commonly used technique to visualize molecular surfaces, such as SES, is by means of triangular meshes [SOS96, TA96, VBJ*94]. However, this method requires the surface triangulation which yields aliasing problems. A more advanced technique is ray-casting the patches that compose the surface. This technique was first introduced by Krone et al. [KBE09] and has been later also employed for dynamic molecules by Lindow et al. [LBPH10].

Perception Enhancement. Several techniques have focused on enhancing the perception of molecular structures. Tarini et al. [TCM06] for example, developed a technique to generate silhouettes and halos around the atoms and bonds. Hermosilla et al. [HGVV16] also used halos computed from a 3D data structure to highlight drugs during the visualization of docking simulations. Krone et al. [KBE09] improved the perception of molecular structures through a set of techniques that could be combined together, such as unsharp masking the depth buffer, silhouette rendering, fogging or transparencies. Lawonn et al. [LKEP14] used hatching lines on molecular surfaces to convey their shape. Van der Zwan et al. [vdZLBI11] provided a system to perform smooth transitions between different representation models, being able to select the ones that best communicate the shape of the molecule. Parulek et al. [PJR*14] presented a technique to visualize smooth transitions between sphere-based molecular representations.

Real-time Global Illumination. Simulating the light transport in a scene at interactive frame rates is a difficult task due to the recursive nature of the problem. However, several techniques have been developed to approximate it in real-time. For a complete analysis of these techniques, the reader can refer to [RDGK12]. Within this paragraph, we rather focus on those techniques directly tailored towards molecular visualization.

Staib et al. [SGG15] used a grid-based approach inspired by Crassin et al. [CNS*11], to simulate global illumination in scenes composed of spheres. Later on, Skånberg et al. [SVGR16] have proposed a regression analysis approach, which has been inspired by the work of Ren et al. [RWG*13]. Due to the inherent problems of regression analysis, with respect to the dimensionality of parameter space to conquer, their approach can only be applied to spherical shapes, in a similar fashion as the original technique presented by Ren et al. was applicable to triangles, only. Moreover, the modulation of the contributions from different objects led to incorrect radiance estimations, i. e. two contributions resulting in red

and blue colors would yield almost zero incoming radiance when using modulation to estimate the total contribution.

Besides the simulation of these Radiosity-like effects, ambient occlusion was used by several authors to improve the shape perception of molecular structures. Tarini et al. [TCM06] computed the visibility of each atom and bond in a pre-process and stored the ambient occlusion terms for each object in a texture atlas. Later, in 2012, Grottel et al. [GKSE12] presented an interactive technique limited to scenes composed of spheres. They computed a coarse approximation of the scene and used it to compute a rough estimation of the occlusion in each pixel. This technique led to two different approaches that proposed a similar extension of the method. Staib et al. [SGG15], limiting the technique to spheres, and Hermosilla et al. [HGVV16], limiting the technique to spheres and cylinders, used a hierarchical data structure to approximate the scene. Then, for each pixel, they computed the ambient occlusion term using Voxel Cone Tracing [CNS*11]. An entirely different approach was presented by Skånberg et al. [SVGR16], which approximated the occlusion generated by a sphere by estimating its projected solid angle. While all these techniques, allow for the incorporation of advanced illumination effects in molecular representations, they are all tailored towards a specific visual representation. To our knowledge, no general molecular illumination models exist, which can be applied to multiple representations.

3. Modeling Diffuse Light Interactions

To enable regression analysis for the derivation of an analytical illumination model, light interactions first need to be formalized. In this section, we describe how we formalize diffuse light interactions and occlusions, which shall both form an integral part of the formalized illumination model.

The amount of light reflected by a point is measured by the reflected radiance. It becomes possible to define this radiance as the sum of the radiance reflected by the surrounding objects plus the radiance arriving from the environment, as the rendering Equation [Kaj86] defines the outgoing radiance from a surface at a given direction as a function of the incoming light from all directions within the hemisphere defined by the normal at each point:

$$L_o(p, \omega_o) = \int_{\Omega} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos \theta_i d\omega_i, \quad (1)$$

where $L_o(p, \omega_o)$ is the outgoing radiance at point p in direction ω_o , $f(p, \omega_o, \omega_i)$ is the object's Bidirectional Reflectance Distribution Function (BRDF) and $L_i(p, \omega_i)$ is the incoming radiance from direction ω_i . However, for perfectly diffuse objects the same amount of light is reflected in all directions, and, therefore, the BRDF becomes constant (c) and can be moved outside the integral.

$$L_o(p, \omega_o) = \frac{c}{\pi} \int_{\Omega} L_i(p, \omega_i) \cos \theta_i d\omega_i \quad (2)$$

Since $\int_{\Omega} \cos \theta_i = \pi$, the albedo c of the object is divided by π .

Now, for explanatory purposes, let's assume we have a scene composed of two convex objects (A and B) separated by a certain distance (Figure 2). Assuming that the only light source in this scene is an environmental radiance of L_a , the radiance reflected by

object A at a given point p in direction ω_o can be computed by the sum of two integrals:

$$L_{Ao}(p, \omega_o) = c \left(\frac{1}{\pi} \int_{\Omega} V_B(p, \omega_i) L_i(p, \omega_i) \cos \theta_i d\omega_i + \frac{1}{\pi} \int_{\Omega} (1 - V_B(p, \omega_i)) L_a \cos \theta_i d\omega_i \right) \quad (3)$$

The first integral computes the radiance arriving at point p (on the surface of the object A) bounced at the surface of object B . Here the incoming radiance for a given direction ω_i (L_i) is modulated by the visibility function $V_B(p, \omega_i)$, which is a binary function that is evaluated to 1 when a ray defined by (p, ω_i) has its first intersection with object B :

$$V_B(p, \omega_i) = \begin{cases} 1 & p \text{ occluded by object } B \text{ in direction } \omega_i \\ 0 & \text{otherwise} \end{cases}$$

Using the second integral we can calculate the radiance arriving at the point p from the environment. Now let's assume that we have a scene similar to the one defined previously, in which we have a convex object A and n convex objects $B_1 \dots B_n$. Using the same procedure as before, we can define the radiance reflected by object A as a summation of the radiance that reaches p reflected by each individual object B_j plus the radiance arriving from the environment:

$$\begin{aligned} L_{Ao}(p, \omega_o) &= c \left(\frac{1}{\pi} \int_{\Omega} V_{B_1}(p, \omega_i) L_i(p, \omega_i) \cos \theta_i d\omega_i + \dots \right. \\ &\quad \left. \frac{1}{\pi} \int_{\Omega} V_{B_n}(p, \omega_i) L_i(p, \omega_i) \cos \theta_i d\omega_i + \right. \\ &\quad \left. \frac{1}{\pi} \int_{\Omega} \left(1 - \sum_{j=1}^n V_{B_j}(p, \omega_i) \right) L_a \cos \theta_i d\omega_i \right) \\ &= c \left(\sum_{j=1}^n f_{B_j \rightarrow A}(p) + L_a \left(1 - \sum_{j=1}^n O_{B_j \rightarrow A}(p) \right) \right) \quad (4) \end{aligned}$$

where $f_{B_j \rightarrow A}(p)$ is a function which evaluates the radiance reflected by object B_j onto point p , and $O_{B_j \rightarrow A}(p)$ describes the occlusion generated by object B_j at point p . $O_{B_j \rightarrow A}(p)$ is in the range $[0, 1]$ and evaluates to 1 when object B_j covers the entire hemisphere and 0 when object B_j does not generate any occlusion wrt. point p .

With the previous derivation, our model defines the radiance at any point on the scene as a summation of the radiance arriving from the objects in its vicinity plus the environmental radiance modulated by the visibility of these objects. Since this model does not take into account self-occlusions, it is limited to scenes composed of convex objects.

4. Sampling Strategies

Equation 4 formalizes the computation of the radiance at a given point. However, $f_{B \rightarrow A}(p)$ and $O_{B \rightarrow A}(p)$ are complicated functions which can be hard to evaluate. For simple objects, such as

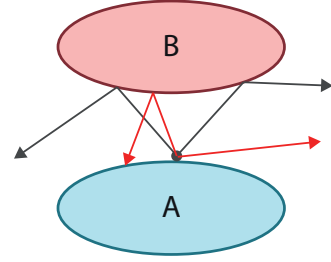


Figure 2: The diffuse interaction between two objects, A and B , is defined as the environmental light that reaches our point of interest from both, the environment and reflected at object B (grey rays).

spheres, these functions can be easily approximated by regression analysis since they are described by a small number of parameters. In contrast, when dealing with more complex objects, a more sophisticated strategy has to be used in order to sample the parameter space efficiently, as the number of samples required by the symbolic regression algorithm increases exponentially with the number of parameters used to describe the new shapes. In this section, we describe strategies that are able to obtain an analytical solution to $f_{B \rightarrow A}(p)$ using symbolic regression even for complex objects defined by several parameters. The key idea is to reduce the number of required samples by selecting those which are more representative.

4.1. Determining the Number of Combinations

Let us assume, that $f_{B \rightarrow A}(p)$ describes the radiance reflected by object B into a point p on the surface of another object A . In order to simplify its computation, we define a different equation $f_{B \rightarrow A}(p)$ for each pair of types of objects B and A . Although, this procedure results in conditional branching, it also reduces the complexity of the function to be approximated. Hence, we initially determine the number of equations to be approximated. For n different types of objects, the number of required equations is n^2 since we have to cover all possible configurations. Once we have identified them, the following simplifications are carried out for each of the cases.

4.2. Parameterize all Possible Spatial Configurations

For each case, we define a set of parameters which identify a unique spatial configuration between two objects of the types studied. The complexity of the regression problem is determined by the number of parameters selected at this step since they determine the size of the space to sample and the complexity of the resulting function. In order to reduce the number of parameters we use the center of one object as the origin of coordinates and define the other with its relative coordinates.

4.3. Parameter Space Sampling

As a next step, we determine the spatial configurations we will use as input to the regression analysis. The accuracy of the resulting function will depend on the number and variety of the configurations we provide.

A straightforward solution would be to consider uniformly distributed samples in the parameter space by varying the value of

one parameter each time. Ensuring that we sample each parameter with enough resolution would allow the symbolic regression algorithm to capture the behavior of our radiance function. However, the number of samples required by this approach increases exponentially with the number of parameters. Assuming that we only take 20 samples for each parameter, for spatial configurations defined by 6 parameters we would generate 64 million of samples, which is infeasible with a limited amount of resources and time, and the main reason why previous approaches have been restricted to single shapes, e.g., [RWG*13, SVGR16].

Therefore, instead of taking uniformly distributed samples, we limit our sampling to those configurations which appear more often in our scenes. First, we discretize the domain of each parameter, limiting them to k different values. Then, we analyze different real-world scenes in order to determine which spatial configurations are more frequent. Based on this analysis, we determine for each pair of objects the values of the different parameters. The discretization of the different domains allows us to identify each configuration by a vector of m integers, $r_{id} = [r_1, \dots, r_m]$, where m is the number of parameters. The components of this vector can be understood as indexes to the array of k different values of each parameter. After analyzing the scenes corresponding to these parameters, we are able to obtain a histogram of these identifiers, providing information of how frequent a spatial configuration took place within the scenes.

In order to obtain a set of configurations with appropriate pairwise differences, we perform another step that reduces the number of spatial configurations even more. This step aims to remove configurations which are very similar, the ones that only differ in one or two parameters and whose values are very close. As a measure of the difference between two configurations, we use the L^2 distance between their identifiers, $d = |r_{id} - r'_{id}|$. Using an iterative algorithm, we select a subset of configurations in which all of them are at a distance d_{min} or more from the rest of the configurations in the subset. This algorithm selects in each iteration a new configuration which meets this requirement. Therefore, d_{min} should be selected experimentally, such that the resulting subset of configurations is small enough (see Section 6.1). Lastly, from the configurations obtained, we select the ones that appear more often in the analyzed scenes until we achieve a small number of configurations which are able to represent the light interactions between the studied objects.

4.4. Samples Reparameterization

Although regression analysis is a powerful tool able to provide an analytical model from a reduced set of samples, it can also produce models with high error. If the function we want to approximate is too complex or the number of parameters of the function is too large, the solution space is too big and, therefore, more difficult to explore. This can result in a model which does not fit well the input data. That was the case of our initial parametrization. Although it helped us to easily determine a set of representative spatial configurations, the result of the regression analysis did not take into account relevant parameters, which lead to incorrect shading. In order to solve this problem, we derive a new parametrization of the problem more closely related to the function we wanted to approximate, guiding thus the algorithm to our desired solution.

The function we intend to approximate, $f_{B \rightarrow A}$, is an integral

over the hemisphere defined by the normal at each point. Our initial parametrization, on the other hand, described the displacement and rotation we should apply to object A to align it to object B . For our new parametrization, we define instead object B by its spherical coordinates with respect to each point on the surface of object A . This parametrization not only provides more information about the function we want to obtain but also introduces more variation in the samples. For each point on the surface of object A , object B is described by different values since its projection is also different.

5. Regression Analysis

In this section we describe the regression analysis we use to approximate our functions. First, for the selected configurations, we execute a path tracer algorithm on a simple scene composed of two objects (as the scene illustrated in Figure 2). We use an environmental radiance (L_a) of 1.0 and a reflectance function for both objects equal to 0.9. We selected this value for the reflectance function since objects with this reflectance function illustrate light absorption whilst reflecting almost all the received light. Then, during the interactive visualization, we derive $f_{B \rightarrow A}(p)$ for different reflectance functions in real-time (see Section 6.3). Moreover, for simplicity, we only simulate a single bounce of the light (grey rays in Figure 2). However, one bounce is enough to plausibly approximate global illumination effects such as diffuse interreflection.

The path tracer computes the following integral for 128^2 different points on the surface of object A using all the spatial configurations selected in the previous step:

$$f_{B \rightarrow A}(p) = \frac{1}{\pi} \int_{\Omega} V_B(p, \omega_i) L_i(p, \omega_i) \cos \theta_i d\omega_i \quad (5)$$

This algorithm generates an image of 128^2 pixels for each spatial configuration. Each pixel of this image encodes the environmental radiance reflected by object B that reaches a given point on the surface of A (grey rays on Figure 2). All the pixels of the images are then used as input to a symbolic regression algorithm.

We use Eureqa [SL09], a symbolic regression package by Nutonian, to approximate our functions. We run the symbolic regression algorithm using the images obtained from the path tracer for all the selected spatial configurations. For each pixel, we compute the reparameterization of object B described in Section 4.4. These values, together with the parameters describing the position of the pixel on the surface of object A , are used as the parameters of the functions the symbolic regression algorithm has to approximate. With this strategy, we are able to obtain equations which fit our input data with very low error.

6. Molecular Illumination Model

In this section we illustrate how, by using the strategies proposed in the previous sections, we are able to obtain different equations describing the diffuse interaction between spheres and cylinders. Then, using these equations, we are able to define a general illumination model for molecular structures.

6.1. Parameter Space

First, we select, by performing the 4 steps of the sampling strategy described in Section 4, the set of representative samples:

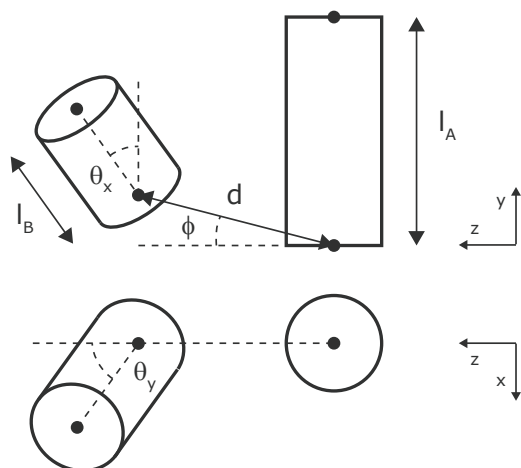


Figure 3: Cylinder-to-cylinder parameterization. The relative position of a cylinder with respect to another can be determined by the use of 6 parameters.

1- Determining the Number of Combinations. As described before, our scenes are only composed of two types of objects, spheres and cylinders. Thus, we need to differentiate 4 possible cases:

Case 1: Object A and B are cylinders.

Case 2: Object A is a cylinder and object B is a sphere.

Case 3: Object A is a sphere and object B is a cylinder.

Case 4: Object A and B are spheres.

2- Parameterize Spatial Configurations. For each of the cases, we define the parametrization of their spatial configurations. The units used for all the parameters are Ångström (Å) for distances and radians for angles.

Case 1: The configuration of a scene composed of two cylinders

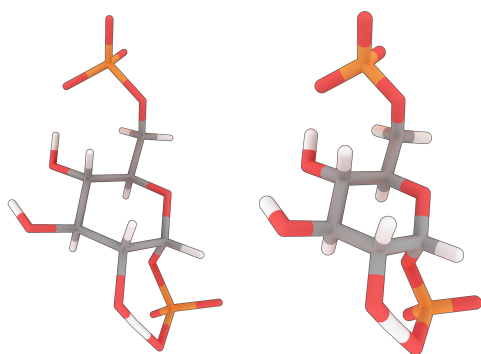


Figure 4: Although during regression analysis we used a fixed radius to define cylinders, our analytical solution to the diffuse inter-reflection can be scaled to approximate the diffuse interreflection between cylinders with other radius. In the left figure we scaled by 0.8 the result of our function to simulate a thinner Licorice model.

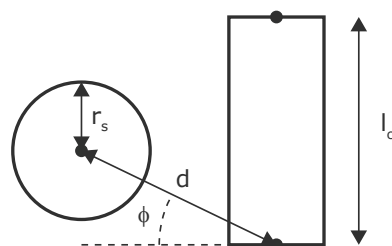


Figure 5: Parameterization of the cylinder-sphere spatial configuration. We need up to 4 parameters to define the relative position of a sphere with respect to a cylinder.

can be fully described by 8 parameters (see Figure 3): the length of both cylinders (l_A and l_B), the radius of both cylinders (r_A and r_B), the distance between the closest extreme points (d), the angle between cylinders in axis X (θ_x), the angle between cylinders in axis Y (θ_y), and the angle formed by the vector between the closest extreme points and the Z-axis (ϕ). This number can be reduced to 6 if we assume a constant cylinder radius. Since there is no standard value for the radius of such elements, we use 0.2 Å for both Balls-and-Sticks and Licorice models. However, if other values need to be considered, the resulting function can be scaled to simulate diffuse interreflections between objects of such radius (see Figure 4).

Cases 2 and 3: A scene composed of a cylinder and a sphere can be defined by 4 parameters (see Figure 5). The length of the cylinder (l_c), the radius of the sphere (r_s), the distance between the center of the sphere and the closest extreme of the cylinder (d), and the angle between the Z-axis and the vector that moves the center of the sphere to the closest extreme of the cylinder (ϕ).

Case 4: A scene composed of two spheres, can be easily described by 2 parameters: the ratio of radius between the two spheres and the distance between them.

3- Parameter Space Sampling. In this step, first, we analyze real-world molecular simulations involving different proteins. We have analyzed 6 simulations composed of thousands of steps each one. From this analysis, we obtained a histogram of the spatial configurations. Then we apply the iterative algorithm to select the most representative ones using $d_{min} = 5$.

Case 1: In this case, we obtained a bit more than 37,000 possible configurations after analyzing the molecular simulations. Then, we reduced them to 576 after our iterative algorithm. Finally, we selected the 201 configurations which appear more than 100 times in the simulations, accounting for 99.95% of the appearances of the 576 configurations.

Cases 2 and 3: In these cases we obtained around 20,000 possible configurations from the analysis of our data. After reducing them to the most representative ones, we obtained 501 configurations. As before, we selected the ones that appear more than 100 times in our simulations, yielding a total of 286 configurations that account for 99.99% of the appearances of the 501 configurations.

Case 4: This case did not need sampling reduction since only 2 parameters needed to be sampled.

4- Samples Reparameterization. In our tests, the resulting equation obtained using regression analysis for the case 1 did not cap-

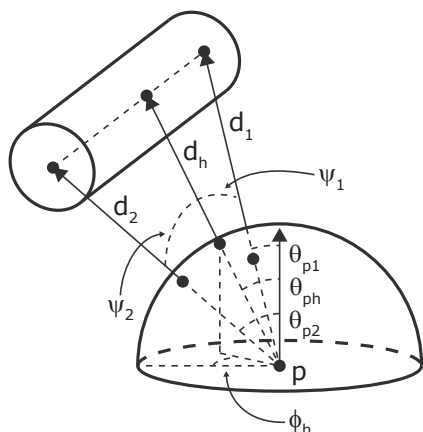


Figure 6: We reparameterize our samples, describing the cylinders by their spherical coordinates with respect to our sampling points. By this reparameterization we are able to introduce more variability in our values (for given cylinder, its spherical coordinates are different for each sample) and introduce information about the problem we want to solve into the regression analysis (our function is basically an integral over the hemisphere defined at each point).

ture well the diffuse interaction between cylinders. The resulting function did not use some of the input parameters, such as θ_y , resulting in wrong interreflections for most of the configurations where θ_y varied. Therefore, we performed the reparameterization described in Section 4.4.

Cases 1 and 3: The reparameterization used when object B is a cylinder is illustrated by Figure 6. First, we project the extremes of cylinder B onto the hemisphere defined at the point p . For each projected point we obtain two parameters, the distance from the extreme to p (d_1 and d_2) and the angle between the projected points and the normal (θ_{p1} and θ_{p2}). Then, from the arc defined by the cylinder in the hemisphere, we compute the closest point to the center of the hemisphere, p_h . From this point we obtain 3 more parameters, its spherical coordinates (d_h , θ_h and ϕ_h). Then, two more parameters are computed which complete the definition of cylinder B , the angle of the arc between the two points of the cylin-

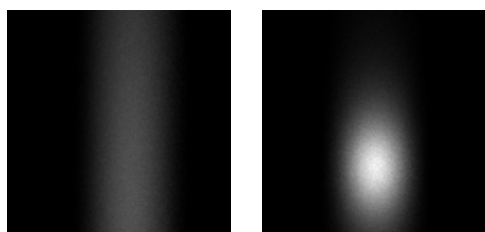


Figure 7: In order to compute the value of the radiance for our sampling points, we used a path-tracing algorithm. Each pixel on the left image presents the radiance reaching different points at the surface of a cylinder reflected by another cylinder. The right image, on the other hand, presents the radiance that reaches different points at the surface of a cylinder reflected by a sphere.

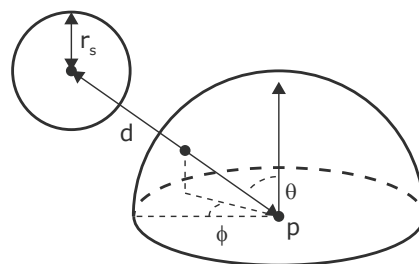


Figure 8: As with the case of cylinder-cylinder interaction, we parameterize the position of a sphere relative to a point using spherical coordinates. Using this parametrization we are able to obtain better approximations with the symbolic regression algorithm.

der B and the point p_h : ψ_1 and ψ_2 . The number of parameters required to define the relative position of a cylinder B from a point p is now 9. Despite increasing the number of parameters, with this new parametrization we are able to obtain an analytical solution for the diffuse interreflection problem which fits our input data well. Note that the proposed parametrization can be ambiguous because the same values can define two different configurations. Nonetheless, these configurations are symmetric which will have almost the same diffuse interaction.

Cases 2 and 4: We reparameterized the cases when object B is a sphere in a similar way we parameterized the data in case 1 and 3 (see Figure 8). For each point on the surface of object A , we project sphere B into the hemisphere defined by the normal of the point, obtaining thus the following parameters: the radius of the sphere (r_s), and the polar coordinates of its center (d , θ and ϕ).

6.2. Regression Results

For the selected configurations we generate the samples using a path tracer algorithm. This algorithm takes less than 30 seconds to generate one of these images, however, this is a pre-process that is executed only once. Figure 7 presents some results obtained by the path tracer algorithm for cases 1 and 2.

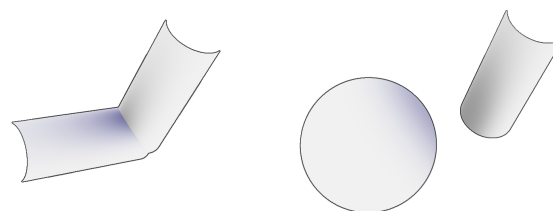


Figure 9: These images present the evaluation in real-time of the functions obtained by the regression analysis to simulate the diffuse interaction between spheres and cylinders. The objects are rendered without color, to illustrate the diffuse interactions they receive. These are calculated assuming the vertical cylinder is blue, while the horizontal one and the sphere are white.

Lastly, we execute the symbolic regression algorithm, obtaining thus the final equations (see Figure 9).

Cases 1 and 3: We used the samples generated for case 1 as input to the symbolic regression algorithm. After 15 hours, we obtained a function which approximates our original input data, with values ranging from 0 to 1, with a mean squared error of 0.000185:

$$f_{B \rightarrow A}(\psi_1, \psi_2, d_2, d_h, \theta_{p_1}, \theta_{p_2}) = 1.25 \exp \left(\left(\frac{\psi_1 + 0.872\psi_2}{1.19 + d_2 + (d_h \theta_{p_1} \theta_{p_2} \theta_{p_1}^{1.19})^{\theta_{p_2}}} \right)^{1.25} \right) - 1.25 \quad (6)$$

Note that this equation does not take into account any parameter defining the position in the surface of cylinder A, i. e. the symbolic regression algorithm approximated the occlusion generated by object A over the surface of object B with a constant. Therefore, we used this same equation to compute the radiance in case 3, where object A is a sphere.

Cases 2 and 4: We also executed the symbolic regression algorithm for the samples generated for case 2. After computing for 17 hours, we obtained a function which approximates our original input data, with values ranging from 0 to 1, with a mean squared error of 0.00034:

$$f_{B \rightarrow A}(r_s, d, \theta) = \text{atan} \left(\frac{34.3r_s^7}{d^7 \exp(\theta^2)} + \frac{5.35r_s^2}{d^2 \exp(\theta^2)} \right) - 0.00196 \quad (7)$$

Again, the obtained function does not depend on object A. Therefore, we used this equation to compute the radiance in case 4 too. Skånberg et al. [SVGR16] also proposed an analytical solution for case 4. However, their formula computes the diffuse interreflection between spheres plus the environmental radiance. This makes it not suitable for our illumination model.

6.3. Model Instance

The equations obtained using our sampling reduction techniques, equations 6 and 7, allow us to define an illumination model following Equation 4. In the following paragraphs, we describe how we derived our illumination model for molecules:

Occlusion Estimation. Equation 4 uses functions $f_{B \rightarrow A}(p)$ to compute the diffuse interactions between objects. However, Equation 4 also relies on the occlusion generated by neighboring objects, $O_{B \rightarrow A}(p)$, to determine the amount of environmental radiance that reaches a point. $O_{B \rightarrow A}(p)$ could be also approximated using regression analysis as $f_{B \rightarrow A}(p)$. In this paper instead, we approximated $O_{B \rightarrow A}(p)$ by scaling the result of equation $f_{B \rightarrow A}(p)$.

From the definition of $f_{B \rightarrow A}(p)$ (Equation 5) we know that the incoming radiance, L_i , will only contribute to $f_{B \rightarrow A}(p)$ when it is reflected by object B. Therefore, the incoming radiance at a point p on the surface of object A for a given direction ω_i is equal to the outgoing radiance from a point p' on the surface of object B in the same direction ω_i . Assuming only one bounce of the light we can define L_i as:

$$\begin{aligned} L_i(p, \omega_i) &= L_{Bo}(p', \omega_i) \\ &= \frac{c_B}{\pi} \int_{\Omega} (1 - V(p', \omega_m)) L_a \cos \theta_m d\omega_m \\ &= c_B L_a \left(\frac{1}{\pi} \int_{\Omega} (1 - V(p', \omega_m)) \cos \theta_m d\omega_m \right) \\ &= c_B L_a AO(p') \end{aligned} \quad (8)$$

where c_B is the reflectance function of object B, L_a is the environmental radiance, and $AO(p')$ is the ambient occlusion term at point p' . Following the approximation carried out by the symbolic regression algorithm, we also consider the occlusion generated by object A constant over the surface of object B. This assumption allows us to approximate $AO(p')$ by a constant, k_o , and therefore derive $O_{B \rightarrow A}(p)$ from $f_{B \rightarrow A}(p)$:

$$\begin{aligned} f_{B \rightarrow A}(p) &= \frac{1}{\pi} \int_{\Omega} V_B(p, \omega_i) L_i \cos \theta_i d\omega_i \\ &= \frac{1}{\pi} \int_{\Omega} V_B(p, \omega_i) c_B L_a k_o \cos \theta_i d\omega_i \\ &= c_B L_a k_o \frac{1}{\pi} \int_{\Omega} V_B(p, \omega_i) \cos \theta_i d\omega_i \\ &= c_B L_a k_o O_{B \rightarrow A}(p) \\ O_{B \rightarrow A}(p) &= \frac{f_{B \rightarrow A}(p)}{c_B L_a k_o} \end{aligned} \quad (9)$$

This approximation of the occlusion generated by an object is able to provide realistic results as shown in Figure 10. However, if a more exact solution is required the occlusion can be computed using other methods, such as ambient occlusion algorithms (see the algorithm proposed by Shanmugan et al. [SA07]) or finding an analytical solution by using regression analysis.

Supporting Different Reflectance Functions. During the computation of Equation 6 and Equation 7 we assumed that objects have a constant reflectance function equal to 0.9. Nevertheless, this is not the case for real scenes. In order to support different reflectance functions, we perform simple manipulations on Equation 5, which allow us to define the radiance reflected by an object B' with a reflectance function $c_{B'}$ by equation $f_{B \rightarrow A}(p)$ defined for an object B with a reflectance function of $c_B = 0.9$. Note that $f_{B' \rightarrow A}(p)$ can be obtained from $f_{B \rightarrow A}(p)$ multiplying by $c_{B'}$ and dividing by c_B , since the outgoing radiance from object B' , can be derived in the same way from the outgoing radiance defined for object B:

$$\begin{aligned} f_{B' \rightarrow A}(p) &= \frac{f_{B \rightarrow A}(p) c_{B'}}{c_B} \\ L_{B'o}(p', \omega_i) &= \frac{L_{Bo}(p', \omega_i) c_{B'}}{c_B} \\ &= \frac{c_{B'} c_B}{\pi c_B} \int_{\Omega} (1 - V_A(p', \omega_j)) L_a \cos \theta_j d\omega_j \\ &= \frac{c_{B'}}{\pi} \int_{\Omega} (1 - V_A(p', \omega_j)) L_a \cos \theta_j d\omega_j \end{aligned} \quad (10)$$

Model Parameters. Our molecular illumination model only has one parameter, k_o , which controls the occlusion generated by the objects of the scene. However, in order to provide control over the

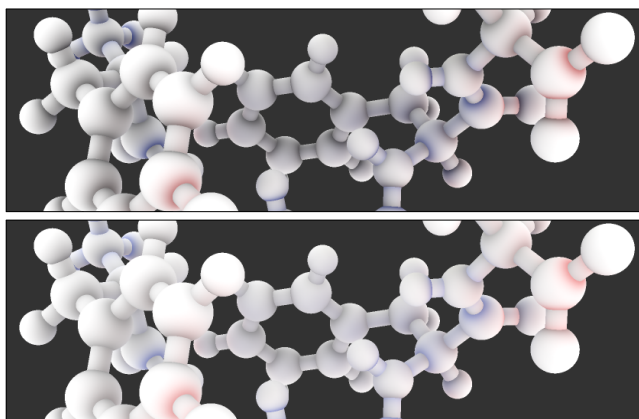


Figure 10: These images illustrate the effect of the parameter k_o , used to approximate the occlusion generated by the nearby objects. The top image was obtained using $k_o = 0.75$ whilst the bottom one was obtained by $k_o = 0.95$. Note that, since k_o divides $f_{B \rightarrow A}$, lower values of this parameter generates higher occlusions.

intensity of the diffuse interactions, we also incorporated a scaling, s_f , and a power, p_f , factors that are applied to the functions $f_{B_j \rightarrow A}(p)$. The effects of these parameters on the visualization are illustrated by Figure 11 and Figure 10.

Supported Representations. This model is used to simulate diffuse interreflections in molecular scenes represented by the most commonly used atomic-based representation methods. Since, the Space-filling, Balls-and-Sticks and Licorice models are representations based on only two geometric primitives, spheres and cylinders, our illumination model can be directly applied here to simulate global illumination effects. The system was developed for cylinders and spheres, however, with some approximations, we can make it work also for SES (see Figure 12): Since SESs are derived from the Space-filling model, for each point in the SES, we compute their final color by using our illumination model as if the scene was composed only of spheres.

The structure of the scenes is very different for each of these representations. Whilst the SF and SES models have a high spatial density, generating a high number of occlusions, the B&S and Licorice models create more sparse visualizations. In order to adapt our functions to these different representations, we adjusted the parameters of our model for each of the representations. The values used for the images of the paper were:

- **Space-filling:** $s_f = 0.2$, $p_f = 0.9$, and $k_o = 0.9$.
- **Balls-and-Sticks:** $s_f = 0.5$, $p_f = 0.95$, and $k_o = 0.85$.
- **Licorice:** $s_f = 2.0$, $p_f = 0.95$, and $k_o = 1.0$.
- **SES:** $s_f = 0.55$, $p_f = 1.15$, and $k_o = 1.05$.

With these values, we are able to obtain realistic results like the ones shown in Figure 1.

7. Implementation

We described an illumination model used to visualize molecules. In this section, instead, we describe some implementation details which allowed us to use this model in real-time.

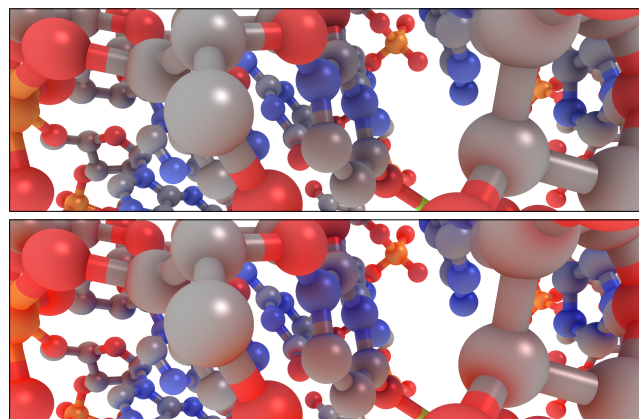


Figure 11: Parameter s_f is used to modulate the diffuse inter-reflection. By increasing this parameter we are able to exaggerate the color bleeding effect. Top image was obtained using $s_f = 0.5$ whilst the bottom image used a $s_f = 1.0$.

7.1. Acceleration Data Structures

Evaluating the diffuse interaction for every pair of objects in the scene can be prohibitive for medium-sized molecules. However, since the sizes of the objects in our scenes are in a well-defined range, we can discard those objects that are farther than a certain distance. Objects far away will have a small projected solid angle and thus, a small diffuse interaction with the point of interest. Hence, to compute the radiance reaching a point, we only have to compute the diffuse interactions with the objects in its vicinity. Determining these objects can be efficiently implemented by using spatial partitioning.

Efficient retrieval of neighboring atoms within a fixed radius is a common problem that is usually solved by using spatial subdivision data structures [Gre07, BSC15, Hoe14]. We adopted the method proposed by Green [Gre07], which was later used in different molecular visualization techniques [KGE11, SVGR16, HKG*17]. We create two data structures, one for atoms and other for bonds, in which we distribute our geometric primitives. These two data structures are computed in milliseconds by the GPU each time the molecular structure is changed. This allows us to maintain real-time interaction during the inspection of full trajectories.

In order to achieve a correct simulation of the diffuse interactions, the cell size needs to be accurately selected. A cell size too small could not take into account, during the retrieval, objects that would generate a significant diffuse interaction with our point of interest. The size of the cells, then, has to ensure that for a given point all the objects of interest are selected whilst only the grid cell in which the point is located and the 26 surrounding cells are queried. This can be achieved by determining the distance at which diffuse interaction is negligible. After trying different spatial configurations in our path tracer, we determined that for a projected solid angle, ω , smaller than 0.04 the diffuse contribution of such objects is no longer relevant. From the equations which define the projected solid angle of our geometric shapes ([Raj14]), we isolate the distance terms and determine at which distances spheres and cylinders project a solid angle of 0.04. These values are then used

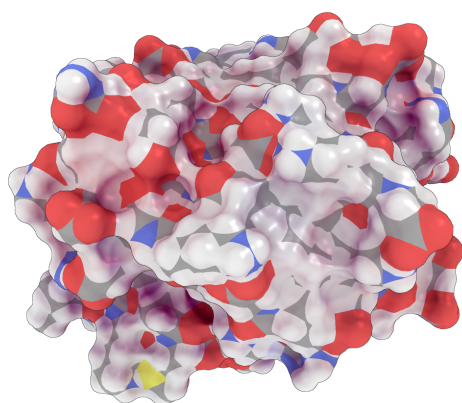


Figure 12: Although solvent excluded surfaces are not formed by spheres and/or cylinders, the representation is derived from the Space-filling model. Using the same illumination model to compute the reflected light by near structures as if our scene was composed only of spheres works well in these cases.

as the size of our regular grids. The cell sizes used for different scenes and different values of ω can be found in Table 2.

With this method, we ensure that we do not miss any relevant diffuse interaction. However, in low performant hardware, the grid cell sizes can be manually selected, reducing thus the number of objects tested.

7.2. Real-time Evaluation of the Interreflections

Once the acceleration data structures are computed, the following algorithm renders our scenes with the proposed illumination model: First, we render our objects into a frame buffer. We store the depth, normal, and object identifier in different render targets. Then, in a second pass, we fetch this information for each pixel and compute the output radiance of the point. To achieve this, we retrieve the objects in the vicinity of the point from the two data structures and, for each one, we use Equation 6 or Equation 7 (depending on the object type) to compute the radiance reflected by the object. The final radiance is computed adding these values following Equation 4.

7.3. Ambient Occlusion

The illumination model proposed in this paper provides a good approximation of global illumination effects (see Figure 13). However, this model only takes into account objects in the vicinity of the point of interest in its computations. When the molecule we intend to visualize is composed of several thousand atoms, far objects, despite not having a significantly diffuse interaction, can occlude the light in their direction.

In order to compute the occlusion generated by these far objects, we use ambient occlusion. Ambient occlusion approximates indirect illumination by modulating the shading of each point by its visibility. In this paper, we adapted the method proposed by Hermosilla et al. [HGVV16] to only compute distant occlusions. For

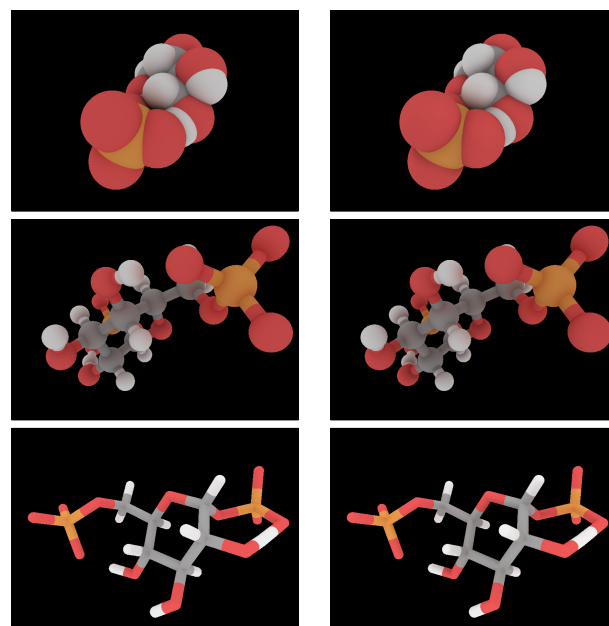


Figure 13: We compared our illumination model (right images) with the results of a path-tracing algorithm (left images). Whilst the Balls-and-Sticks and Licorice models achieve very similar results, the Space-filling model generates less pronounced shadows, though still plausible. And the diffuse interactions appear in the same areas as with the path tracing algorithm.

more information about this technique, the user can refer to the original paper.

8. Results

In this section, we present the results obtained by our algorithm, in terms of image quality and performance achieved. All the measures were taken using a computer with the following configuration: Intel i7 at 3.6 GHz, with 16 Gb of RAM, and a GeForce GTX980, using a screen resolution of 1280×720 pixels.

The illumination model we introduced in this paper works in real-time for molecules up to 45 K atoms. Table 1 presents the results obtained for different scenes. Since the bottleneck of our algorithm is the pixel shader, we provide different configurations for analysis purposes. We rendered the tested molecules making them occupy as much of the screen as possible without being clipped, testing thus the worst case scenario. For all the representations our method supports, we provide the milliseconds required to draw a frame using different cell sizes. The Space-filling and SES representations are able to render molecules up to 45 k atoms at more than 60 frames per second for all the cell sizes tested. On the contrary, the Balls-and-Sticks and Licorice models present a drop in performance for higher cell sizes, since they require to query two data structures. For these models, we measured a performance of around 20 frames per second. However, these cell sizes are bigger than the ones used in real scenarios. Table 2 presents the cell sizes calculated for different values of ω in molecules represented by the Balls-and-Sticks model. For a ω equal to 0.04 (for which we cap-

Table 1: Results obtained for different representation methods. Each column presents the time in milliseconds needed to render a frame using the different representation methods with different cell sizes (in Å) of our acceleration data structures. The first number indicates the cell size of the data structure used to store the atoms and the second one the cell size of the data structure used to store bonds. Since the Space-filling and SES models do not visualize bonds between atoms, only the cell size for the data structure storing the atoms is presented. Notice how we obtain interactive to real-time frame rates for even the largest models evaluated.

Molecule (#atoms / #bonds)	Space-filling			Balls-and-Sticks			Licorice			SES		
	2 / -	4 / -	8 / -	2 / 1	4 / 2	8 / 4	2 / 1	4 / 2	8 / 4	2 / -	4 / -	8 / -
1OGZ (2114 / 2014)	3.723	4.461	7.584	4.811	7.418	20.909	4.656	6.522	17.869	4.376	4.802	8.769
4NKG (4507 / 4596)	3.543	4.169	7.143	4.991	7.878	23.509	4.411	7.029	20.127	4.016	4.671	8.841
1AF6 (10517 / 10485)	4.354	5.602	12.355	7.058	13.151	48.638	6.495	11.510	41.973	5.054	6.782	14.712
5XYU (45157 / 48953)	4.369	5.332	8.525	12.892	14.932	42.522	12.226	13.866	38.339	4.444	5.724	9.828

Table 2: Cell sizes (in Å) of the data structures storing atoms (c_a) and storing bonds (c_b), for the minimum solid angle considered (ω) in the Balls-and-Sticks model. Moreover, we provide the mean occupancy of these cells in parenthesis. In our tests, we found that structures with a projected solid angle of 0.04, we are able to capture all the relevant diffuse interactions. In this table, we provide cell sizes for other values of ω for analysis purposes.

Molecule	$\omega = 0.1$		$\omega = 0.04$		$\omega = 0.01$	
	c_a	c_b	c_a	c_b	c_a	c_b
1OGZ	3.04 (1.86)	1.31 (2.00)	4.79 (4.25)	2.07 (3.02)	9.57 (13.72)	4.13 (7.50)
4NKG	3.04 (1.97)	1.30 (1.94)	4.79 (4.63)	2.05 (2.95)	9.57 (14.73)	4.09 (7.16)
1AF6	3.04 (1.95)	2.01 (2.87)	4.79 (4.97)	3.20 (4.86)	9.57 (16.96)	6.43 (21.22)
5XYU	3.04 (2.04)	2.02 (3.04)	4.79 (4.77)	3.22 (5.29)	9.57 (15.87)	6.47 (21.38)

ture almost all the diffuse interactions) we determined cell sizes of 5 Å (Atoms) and 3 Å (Bonds). As we present in Table 1, similar cell sizes present a performance of more than 60 frames per second in all the representations.

Moreover, we compared our method with the results of a path tracing algorithm. The images in Figure 13 show that for the Balls-and-Sticks and Licorice models, our algorithm almost presents no difference. On the other hand, for the Space-filling model (and therefore SES), the results of our algorithm differ from the ground truth. We believe that this is the result of the occlusion approximation carried out in Section 7. Nevertheless, our technique generates diffuse interreflections in the same areas as the path tracer.

Lastly, we conducted a domain expert interview. All received comments were very positive. However, they found the major improvement in the visualizations of Balls-and-Sticks and Licorice models. They pointed out that usually, with these models, one needs to constantly move the molecule in order to gain a sense of the relative positions. With our illumination model, it is not necessary, since even static images are able to communicate such information.

9. Limitations

The proposed method efficiently approximates diffuse interreflections for molecular models. However, it is not exempt from limitations. Equation 4, describes the radiance arriving at a point on the surface of object A as the summation of individual contributions from the objects in its vicinity, $\{B_j | 0 < i \leq n\}$. Although this model takes into account whether object B_j is partially or completely occluded by another object B_k through the visibility function $V_{B_j}(p, \omega_i)$, our implementation of the model assumes that no occlusions exist between objects B_j and B_k . However, this is not the case for real scenes. The radiance arriving from object B_j to p should be modulated by the part of object B_j that is occluded by objects B_k when is observed from point p . We could apply a similar technique as the one used by Skånberg et al. [SVGR16]. Nevertheless, as they described in the paper, the effect of overbleeding was rarely noticed in the scenes they tested. Since our cell sizes have a reduced size and the diffuse interaction is a local effect that loses strength with the distance, we decided to not apply any weighting factor based on occlusions between different objects.

Moreover, our current implementation is only able to take into account one representation type at a time. In order to combine different representations in the same visualization, extra logic should be added into the fragment shader. The application should maintain a set of data structures for each type of representation. Then, the fragment shader should retrieve the objects in the vicinity from each of these data structures and apply the corresponding values of the parameters s_f , p_f , and k_o .

10. Conclusions

In conclusion, we have introduced a set of sampling strategies, in the context of regression analysis, that are able to reduce the sampling of the parameter space by selecting the most representative samples and reparameterizing them based on the shape of the equation we aim to approximate. Furthermore, we applied these strategies to obtain an analytical solution for the diffuse interaction between spheres and cylinders. In these cases, we reduced the number samples to around 200 images which allowed us to obtain models with a very low error in less than 17 hours. Lastly, we applied these new found equations to define a general illumination model which simulates global illumination effects in real-time for the most commonly used atom-based molecular representations. This model defines the radiance arriving at a point as the summation of the re-

flected light by the objects in the vicinity plus the radiance of the environment modulated by the visibility of these objects.

In the future, we will like to study the applicability of our methods to simulate diffuse interaction in other types of scenes, such as triangular meshes.

Acknowledgements

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under grants RO 3408/2-1 (Prolint) and RO 3408/3-1 (Inviwo), by the Spanish Ministerio de Economía y Competitividad, and by 839 FEDER (EU) funds under the Grants No. TIN2014-52211-C2-840 1-R and TIN2017-88515-C2-1-R.

References

- [BSC15] BEHLEY J., STEINHAGE V., CREMERS A. B.: Efficient radius neighbor search in three-dimensional point clouds. In *2015 IEEE International Conference on Robotics and Automation* (May 2015), pp. 3625–3630. 8
- [CNS*11] CRASSIN C., NEYRET F., SAINZ M., GREEN S., EISEMANN E.: Interactive indirect illumination using voxel cone tracing. *Computer Graphics Forum (Proceedings of Pacific Graphics 2011)* 30 (sep 2011). 3
- [FKE13] FALK M., KRONE M., ERTL T.: Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. *Computer Graphics Forum* (2013), 195–206. 2
- [Fos11] FOSTER D. H.: Color constancy. *Vision Research* 51, 7 (2011), 674 – 700. Vision Research 50th Anniversary Issue: Part 1. 2
- [GKSE12] GROTTTEL S., KRONE M., SCHARNOWSKI K., ERTL T.: Object-space ambient occlusion for molecular dynamics. In *Pacific Visualization Symposium (PacificVis)*, 2012 IEEE (Feb 2012), pp. 209–216. 3
- [Gre07] GREEN S.: *White paper: CUDA Particles*. Tech. rep., 2007. 8
- [Gum03] GUMHOLD S.: Splatting illuminated ellipsoids with depth correction. In *VMV* (2003), Ertl T., (Ed.), Aka GmbH, pp. 245–252. 2
- [HGVV16] HERMOSILLA P., GUALLAR V., VINACUA A., VÁZQUEZ P.: High quality illustrative effects for molecular rendering. *Computers & Graphics* 54 (2016), 113 – 120. Special Issue on CAD/Graphics 2015. 3, 9
- [HKG*17] HERMOSILLA P., KRONE M., GUALLAR V., VÁZQUEZ P.-P., VINACUA A., ROPINSKI T.: Interactive gpu-based generation of solvent-excluded surfaces. *The Visual Computer* 33, 6 (2017), 869–881. 8
- [Hoe14] HOETZLEIN R. C.: Fast fixed-radius nearest neighbors: Interactive million-particle fluids. In *GPU Technology Conference* (2014). 8
- [Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150. 3
- [KBE09] KRONE M., BIDMON K., ERTL T.: Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov 2009), 1391–1398. 3
- [KGE11] KRONE M., GROTTTEL S., ERTL T.: Parallel contour-buildup algorithm for the molecular surface. In *Biological Data Visualization (BioVis)*, 2011 IEEE Symposium on (Oct 2011), pp. 17–22. 8
- [KKF*16] KOZLÍKOVÁ B., KRONE M., FALK M., LINDOW N., BAADEN M., BAUM D., VIOLA I., PARULEK J., HEGE H.-C.: Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum* (2016). 2
- [LBPH10] LINDOW N., BAUM D., PROHASKA S., HEGE H.-C.: Accelerated visualization of dynamic molecular surfaces. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 943–952. 3
- [LKEP14] LAWONN K., KRONE M., ERTL T., PREIM B.: Line integral convolution for real-time illustration of molecular surface shape and salient regions. *Computer Graphics Forum* 33, 3 (2014), 181–190. 3
- [LMPSV14] LE MUZIC M., PARULEK J., STAVRUM A. K., VIOLA I.: Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. *Computer Graphics Forum* 33, 3 (2014), 141–150. 2
- [LJR*14] PARULEK J., JÖNSSON D., ROPINSKI T., BRUCKNER S., YNNERMAN A., VIOLA I.: Continuous levels-of-detail and visual abstraction for seamless molecular visualization. *Computer Graphics Forum* 33, 6 (2014), 276–287. 3
- [Raj14] RAJPOOT H. C.: *Advanced Geometry: Mathematical Analysis of Unified Articles*, 1 ed. Notion Press, 2014. 9
- [RDGK12] RITSCHER T., DACHSBACHER C., GROSCH T., KAUTZ J.: The state of the art in interactive global illumination. 160 – 188. 3
- [RE05] REINA G., ERTL T.: Hardware-Accelerated Glyphs for Mono- and Dipoles in Molecular Dynamics Visualization. In *EUROVIS 2005: Eurographics / IEEE VGTC Symposium on Visualization* (2005), Brodlie K., Duke D., Joy K., (Eds.), The Eurographics Association. 2
- [RWG*13] REN P., WANG J., GONG M., LIN S., TONG X., GUO B.: Global illumination with radiance regression functions. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 130. 2, 3, 4
- [SA07] SHANMUGAM P., ARIKAN O.: Hardware accelerated ambient occlusion techniques on gpus. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2007), I3D '07, ACM, pp. 73–80.
- [SGG15] STAIB J., GROTTTEL S., GUMHOLD S.: Visualization of particle-based data with transparency and ambient occlusion. *Computer Graphics Forum* 34, 3 (2015), 151–160. 2, 3
- [SL09] SCHMIDT M., LIPSON H.: Distilling free-form natural laws from experimental data. *Science* 324, 5923 (2009), 81–85. 2, 5
- [SOS96] SANNER M. F., OLSON A. J., SPEHNER J. C.: Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers* 38, 3 (Mar 1996), 305–320. 3
- [SVGR16] SKÅNBERG R., VÁZQUEZ P., GUALLAR V., ROPINSKI T.: Real-time molecular visualization supporting diffuse interreflections and ambient occlusion. *IEEE transactions on visualization and computer graphics* 22, 1 (01 2016), 718–727. 2, 3, 4, 7, 8
- [SWBG06] SIGG C., WEYRICH T., BOTSCH M., GROSS M.: Gpu-based ray-casting of quadratic surfaces. In *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics* (Aire-la-Ville, Switzerland, Switzerland, 2006), SPBG'06, Eurographics Association, pp. 59–65. 2, 3
- [TA96] TOTROV M., ABAGYAN R.: The contour-buildup algorithm to calculate the analytical molecular surface. *J. Struct. Biol.* 116, 1 (1996), 138–143. 3
- [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 (Sept. 2006), 1237–1244. 2, 3
- [VBJ*94] VARSHNEY A., BROOKS F. P., JR., WILLIAM J., WRIGHT W. V.: Linearly scalable computation of smooth molecular surfaces. In *IEEE Computer Graphics and Applications Vol 14* (1994), p. pp. 3
- [vdZLB11] VAN DER ZWAN M., LUEKS W., BEKKER H., ISENBERG T.: Illustrative molecular visualization with continuous abstraction. *Computer Graphics Forum* 30, 3 (2011), 683–690. 3