Improving Perception of Molecular Surface Visualizations by Incorporating Translucency Effects

P. Hermosilla[1] and S. Maisch[1] and P.-P. Vázquez[2] and T. Ropinski[1]

[1]Ulm University, Germany
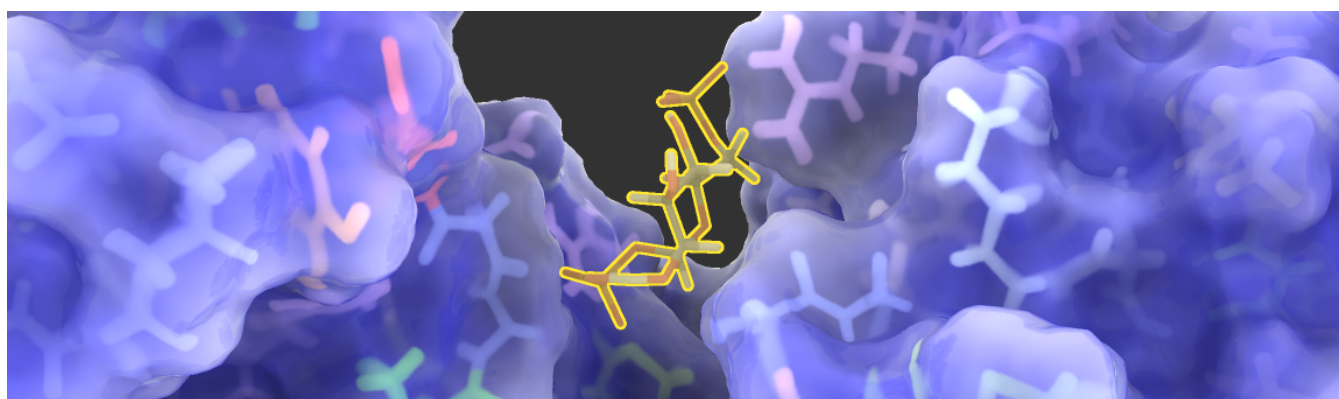[2]Universitat Politècnica de Catalunya, Spain

Figure 1: *Visualization of the docking process of a drug (molecule highlighted by a yellow silhouette) into a protein represented by its solvent excluded surface. In order to improve shape and topology perception of the molecular surface, we approximate the subsurface scattering effect characteristic of translucent objects in real-time. Moreover, we apply transparencies to visualize structures close to the surface, something of key importance when analyzing in detail features of the molecular surface. We further improve the perception of the surface curvature by incorporating reflections and refractions. Lastly, in order to reduce the visual noise, we filter the amino acids by its distance to the drug.*

Abstract

Molecular surfaces are a commonly used representation in the analysis of molecular structures as they provide a compact description of the space occupied by a molecule and its accessibility. However, due to the high abstraction of the atomic data, fine grain features are hard to identify. Moreover, these representations involve a high degree of occlusions, which prevents the identification of internal features and potentially impacts shape perception. In this paper, we present a set of techniques which are inspired by the properties of translucent materials, that have been developed to improve the perception of molecular surfaces: First, we introduce an interactive algorithm to simulate subsurface scattering for molecular surfaces, in order to improve the thickness perception of the molecule. Second, we present a technique to visualize structures just beneath the surface, by still conveying relevant depth information. And lastly, we introduce reflections and refractions into our visualization that improve the shape perception of molecular surfaces. We evaluate the benefits of these methods through crowd-sourced user studies as well as the feedback from several domain experts.

CCS Concepts
●Human-centered computing → Scientific visualization; ●Computing methodologies → Computer graphics; Rendering;

1. Introduction

Molecular surfaces provide a compact description of the space occupied by a molecule and its accessibility. These surfaces are com-

monly used by biochemists for analyzing molecular structures or the results of molecular simulations. Among the different derived surfaces, the Solvent Excluded Surface (SES) [Ric77] is one of the most frequently used. The SES describes the area around a

molecule which is accessible by another small molecule (usually approximated by a sphere).

Despite providing a good overview of the molecular shape, these techniques are not well suited for a complete analysis of a molecular structure. The abstraction these techniques perform of the atomic data prevent the identification of fine grain features of such structures. Moreover, these representations occupy a lot of 3D space, generating high number of occlusions. Due to these occlusions, the internal conformation of the molecule is not visible and the external shape is sometimes difficult to perceive. All these factors, make molecular surfaces only suitable in use cases in which experts are not interested in a detailed analysis of the structure. During the analysis of the results of docking simulations, for example, in which a drug has to dock into a protein (see Figure 1), visualizing the amino acids close to the surface of the docking position is of key importance. With standard visualizations of these surfaces, this is not possible. Domain experts have to alternate between different representations and constantly rotate the point of view to gain insight into these structures.

Some of these problems have been recently addressed by different authors [JPSK16, KKP*13], focusing only on rendering transparent surfaces. Although the use of transparency may partially alleviate the clutter, the presented methods are computationally demanding, difficult to implement and, for medium-sized or larger molecules, they still generate cluttered images (see Figure 2).

In this paper, we introduce a set of techniques designed to address these problems by simulating different visual effects characteristic of translucent materials. Our collection of techniques aims to increase the information presented by a standard SES visualization, improving at the same time the shape perception of the surface. We are able to achieve our goals by providing information of the thickness and curvature of the surface through different photorealistic effects, and reducing clutter when presenting the internal structures of the molecule by selecting only the most relevant. First, we introduce a simple technique which simulates the subsurface scattering (SSS) effect by modulating the color of the surface based on its thickness, providing thus information of the thickness of the surface. Second, we present a new technique to render the structures which are close to the surface, providing extra information about the molecule without generating visual noise. Moreover, we incorporate reflections and refractions, thus improving the shape perception. Lastly, we support filtering of the presented structures based on the distance to a point of interest (such as the drug position in docking simulations).

We evaluated our techniques at different levels. First, we obtained expert feedback by presenting our techniques to five biochemists with different areas of expertise. Furthermore, in order to evaluate the perceptual quality improvement obtained using our methods, we ran two user studies which demonstrate the benefits of our methods over the most commonly used visualization techniques.

Our contributions are summarized as follows:

- We introduce transparency and translucency effects to improve the shape perception in visualizations of molecular surfaces.
- We propose an algorithm to compute subsurface scattering in such representations.
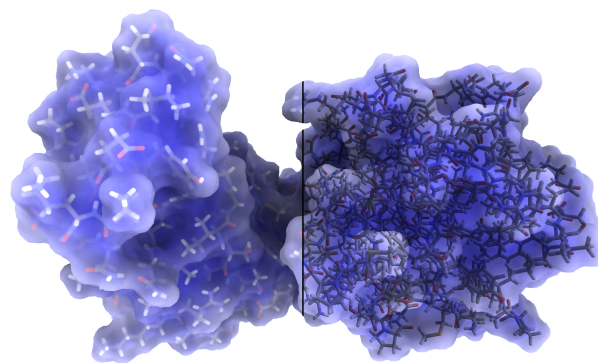

Figure 2: *Comparison of our method to visualize internal structures (**left**) with the transparencies commonly used by state-of-the-art methods (**right**). With our method, only relevant atoms are presented avoiding cluttered images such as the one presented in the right figure.*

- We present a technique to generate transparencies, allowing the visualization of structures beneath to the surface.
- Lastly, we incorporate reflections and refractions in our visualizations, thus improving the curvature perception of such surfaces.

2. Related Work

In this section, we will briefly discuss the related work regarding molecular surface visualization, subsurface scattering, and transparencies.

Visualization of Molecular Surfaces: One of the most commonly used techniques to visualize molecular surfaces is by triangulating the parametric patches that define the surface [SOS96, TA96, VBJ*94] in a pre-process and, then, render the triangular mesh using the standard rasterization algorithm. However, this method suffers from aliasing problems and adds an extra computational cost that can be prohibitive in scenarios where the surface has to be computed interactively. Last decade, Krone et al. [KBE09, KDE10] and Lindow et al. [LBPH10] developed different algorithms which allowed the computation of such surfaces interactively. Due to the interactivity requirement raised by these new methods, new visualization techniques were also developed. For their visualization, both methods rendered the patches of the molecular surface by using ray-marching of the pixel shader. In 2012, Parulek and Viola [PV12] presented a completely new approach which used sphere tracing ([HSK89]) on a set of functions describing the SES. Another approach based on sphere tracing was presented by Hermosilla et al. [HKG*17]. This method was based on the technique presented by Lindow et al. [LBH14] and it used a distance field stored in a mipmapped 3D texture to store the SES which was accessed later during rendering.

Moreover, molecular surfaces have been widely used for communicating color-encoded information. Krone et al. [KBE09, KDE10] encoded in the surface using a color scale the temperature or B-factor of each atom. The same authors also used several

techniques to improve shape perception, such as Depth darkening or silhouettes. Other visualizations present, also by using a color scale, the charge of the different amino acids [HKG*17]. Moreover, some visualization packages, such us Chimera [PGH*04], also provide visualizations where they encode by a color scale information about the amino acid hydrophobicity or the surface electrostatic potential. A completely different approach was proposed by Lawonn et al. [LKEP14], where they used a non-photorealistic rendering technique to enhance the shape perception of such representations.

Subsurface Scattering: Subsurface scattering is a well-researched effect of volumetric objects in computer graphics. Its foundation is described by Radiative Transport Theory as covered in depth by Chandrasekhar [Cha60]. In computer graphics, it was first introduced by Blinn [Bli82] for simulating clouds and dusty surfaces. Single scattering effects for homogeneous participating media were presented by Hanrahan and Krueger [HK93]. Most current simulations of subsurface scattering are based on the BSSRDF approximation which is derived from diffusion theory by Jensen et al. [JMLH01]. This approximation has been improved several times, most recently by D'Eon and Irving [DI11] and Habel et al. [HCJ13] and is also applicable in real-time scenarios as described for example by Jimenez et al. [JZJ*15].

All of the methods previously mentioned for rendering subsurface scattering in real time either rely on triangle meshes for rendering or just do a surface 'blurring' to simulate the effect. In our case we have the SES of a molecule given as a signed distance function that is sphere traced for rendering as described in [HSK89].

Transparencies: Simulating transparent objects has been a widely studied topic in computer graphics. Yang et al. [YHGT10] created per-pixel linked lists in order to achieve order-independent transparency. McGuire and Mara [MM17] proposed a combination of techniques that approximate several effects of transparencies order independently in real-time. Other effects of transparent objects, such as refractions, were successfully simulated in real-time by using a screen space approach. Davis and Wyman [DW07], and Wyman [Wym05] presented different methods based on this idea obtaining realistic approximations of this effect.

Transparencies have been also studied in the context of molecular surfaces since visualizing the interior structures is of key importance during the analysis process of a protein. Some authors [KBE09, LBPH10, LKEP14] simulated transparencies by blending the molecular surface with the structures rendered in the background. Although this is a fast approximation, in these techniques it is difficult to identify the relative position of the objects. Kauker et al. [KKP*13] extended the method of Yang et al. [YHGT10] to compute more realistic transparencies in molecular surfaces. In 2016, Jurčík et al. [JPSK16], also presented a technique to simulate semi-transparent objects aiming at visualizing interior cavities. However, since these methods consider all the interior structures, for large molecules, they can generate cluttered images (see Figure 2).

3. Algorithm

Most natural objects are not completely opaque. Light travels inside them exiting at a different point from where they entered. This effect is known as subsurface scattering. Subsurface scattering (especially the transmission effect) can provide extra information about

the 3-dimensional structure of an object. Some natural materials have a high amount of scattering which generates a blurring effect on the structural information. Since molecular visualizations usually create an abstraction of real molecular structures, there is no natural subsurface scattering effect to simulate. Therefore, for these visualizations, we are able to adjust the effect to low, mostly forward scattering. This also allowed us to show the actual atoms of the molecule directly below the surface (and only those) in a natural way. These atoms are of high interest for experts since they describe the properties of the molecular surface and, therefore, have more influence (due to proximity) in the interaction between proteins. The goal of our techniques is to show the molecular surface in a similar way to dense, backlit fog which lets the viewer get some information about its thickness on one hand, whilst showing at the same time objects that are placed near the surface.

3.1. SES Representation

Some of the algorithms described in this paper are based on a distance field describing the scene. The distance field of an SES can be computed efficiently by grid-based methods [HKG*17]. The main benefit of these methods, when compared to analytical solutions, is their performance. However, big cell sizes can introduce errors in the resulting SES. Therefore, a trade-off between grid size and cell size has to be maintained to generate correct surfaces in a reasonable computation time. Nevertheless, these methods are conservative and they are commonly used only for interactive visualization purposes.

In this work, we used a similar approach as Hermosilla et al. [HKG*17] to compute the SES. They stored the SES as a signed distance field in a 3D texture. Using a two-pass algorithm they were able to compute the distance field in the vicinity of the surface in real time. We used this algorithm to compute our SES as a signed distance field. However, since we are also interested in the value of the SDF inside the surface, we executed a set of additional passes (implemented by a compute shader) to propagate the correct distances to all the grid points of our 3D texture. With this algorithm we are able to update a grid of 128^3 grid points at interactive rates, thus allowing the interactive analysis of molecular simulations. For bigger grid resolutions, although this computation requires more time (in the order of a few seconds), it has to be computed only once for each new simulation step.

Beside supporting SES stored as signed distance fields, our subsurface scattering algorithm can also support the SES representation proposed by Parulek and Viola [PV12], since they use a set of analytic functions describing the distance to the SES. However, for other surface representations other techniques have to be applied. For surfaces represented by a set of patches [KBE09, LBPH10], a hierarchical data structure storing the patches could accelerate the retrieval patches in the vicinity of the point of interest. Then, the distance to the surface can be directly computed by the implicit functions describing the patches. For surfaces represented by triangle meshes [SOS96, TA96, VBJ*94], existing algorithms to compute the signed distance field [ED08, SPG03] can be used to generate this data structure.
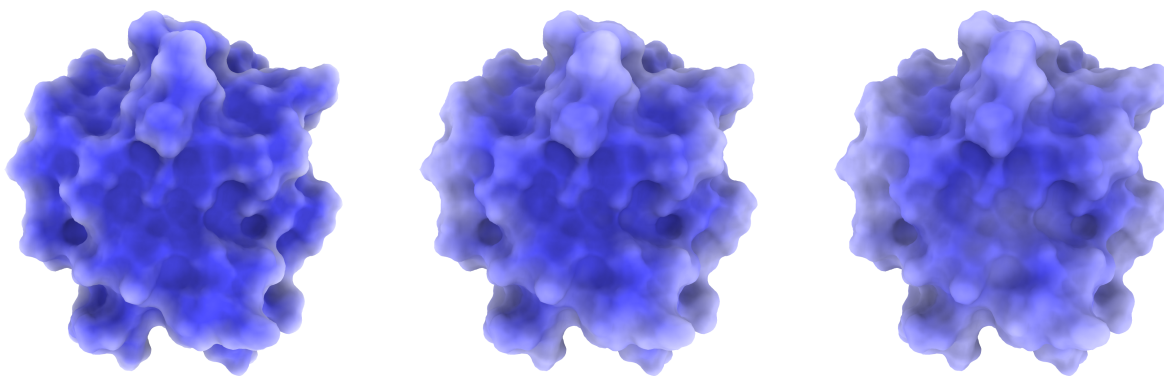
Figure 3: *Our subsurface scattering algorithm (together with ambient occlusion) applied to the same molecule with three different values of l_s. From left to right, $l_s = 3$ Å, $l_s = 5$ Å and $l_s = 7$ Å. Note how the visualization reveals the thinner areas of the surface by encoding them with a brighter color as we increase the value of l_s.*

3.2. Subsurface Scattering

In order to generate the subsurface scattering effect on molecular surfaces, we used an algorithm based on the technique introduced by Evans et al. [Eva06] to compute ambient occlusion. They used sphere tracing to render a scene described by a signed distance function (SDF). Each ray traverses the scene until they intersect with a surface (point $\mathbf{x}$ with $SDF(\mathbf{x}) = 0$). Then, at $\mathbf{x}$, another ray is traced in the direction of the normal. By accumulating the SDF values along this ray they were able to approximate the ambien occlusion at point $\mathbf{x}$. Our algorithm also computes, by using sphere tracing, the intersection of the viewing ray $\vec{r}$ with a molecular surface described by an SDF. This SDF describes the distance from a given point on the space to the molecular surface, begin positive when the point is outside the surface and negative when it is inside. Once the intersection point $\mathbf{x}$ is found, we follow the view ray inside the surface a fixed distance l_s and we evaluate the SDF one more time at point $\mathbf{x}_s$:

$$l = SDF(\mathbf{x_s}) \qquad \text{where} \qquad \mathbf{x_s} = \mathbf{x} + l_s \vec{r} \qquad (1)$$

The result of this evaluation, l, is the distance from the point $\mathbf{x}_s$ inside the molecular surface to the closest point of the surface. This value is then used to approximate the subsurface scattering factor by:

$$s = \frac{l_s - l}{2l_s} \qquad (2)$$

Note that s is in the range $[0, 1]$. It evaluates to 1 when the closest point on the surface to $\mathbf{x}_s$ is our initial intersection point $\mathbf{x}$, and therefore $l = -l_s$ ($\mathbf{x_0}$ on Figure 4). On the contrary, s evaluates to 0 when the intersection point $\mathbf{x}$ lies on a surface with thickness equal to 0. In this case, our point $\mathbf{x}_s$ will lay outside the surface at a distance $l = l_s$ from $\mathbf{x}$ (similar to point $\mathbf{x_2}$ on Figure 4).

Once we compute the value of s for a given pixel, we use it to determine its final color. Instead of using the standard Blinn-Phong

shading to compute the lighting on the surface of SES, we add s to each component of the color of the surface. We decided to do not use Blinn-Phong shading since it could get in conflict with the information we intended to communicate. Specular reflections on the surface generated by this illumination model could be wrongly interpreted as our technique communicating a thin area on the surface. However, our application still provides the option of adding Blinn-Phong shading in order to generate pictures for publication or similar purposes.

Moreover, we incorporate in our technique extra information about the shape of the molecule by using ambient occlusion. In our images, ambient occlusion factors are computed using the algorithm described by Hermosilla et al. [HKG*17].

Figure 3 presents the same molecule rendered using this algorithm with three different values of l_s: 3Å, 5Å and 7Å.

Visual Identification of Cavities: The proposed algorithm not only is able to communicate the thickness of the surface but also
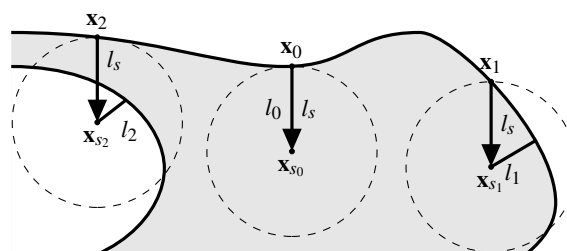

Figure 4: *Three different cases for the evaluation of the distance l used to calculate the subsurface scattering. The arrows show the viewing ray $\vec{r}$ inside the volume surrounded by the SES (grey). The dashed circles have radius l_s and show the area where closest points can be. If the closest point to $\mathbf{x}_{s_0}$ is $\mathbf{x}_0$ then $l_0 = -l_s$ and $s = 1$ as seen in the middle. In the right, $\mathbf{x}_{s_1}$ is inside the the surface and $l_1 \in [-l_s, 0]$ which leads to $s \in [0.5, 1]$. The left example shows $\mathbf{x}_{s_2}$ being inside a cavity. In this case $l_2 \in [0, l_s]$ and $s \in [0, 0.5]$.*
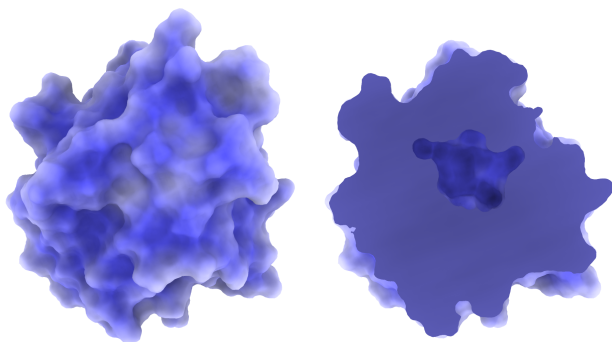
Figure 5: *A molecule with a cavity visualized by our technique. The center of the molecule's SES in the left image is drawn in a brighter color due to the subsurface scattering effect generated by the internal cavity. On the right side, a cut through the same molecule is shown to prove that there is indeed a cavity.*

Figure 7: *The algorithm used to approximate the distinctive blurring of translucent materials. For a central pixel, p_c, we compute its final color by accumulating its value and the values of neighboring pixels, p_1 and p_2, scaling them by a Gaussian function defined for each of them. The width of these Gaussians is computed based on the distance of the structure to the surface, d_c, d_1 and d_2.*

provides information of the internal cavities of the molecular surface. Surface areas close to cavities are interpreted by our algorithm as a thin surface and, therefore, shaded using a whiter color (point $\mathbf{x_2}$ in Figure 4).

The objective of this algorithm is not substitute existing cavity algorithm detection. The state of the art algorithms in this field can provide very accurate results [KKL*16]. Nevertheless, our method can be used to interactively visualize ongoing simulations and provide the first hint about these cavities (see Figure 5). Later, during a more deep analysis carried out offline, the simulation can be analyzed by a more accurate algorithm and all the cavities can be extracted.
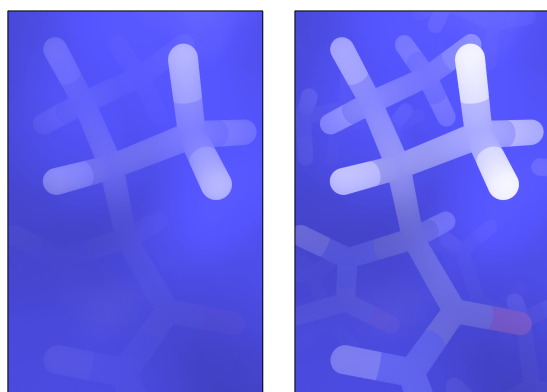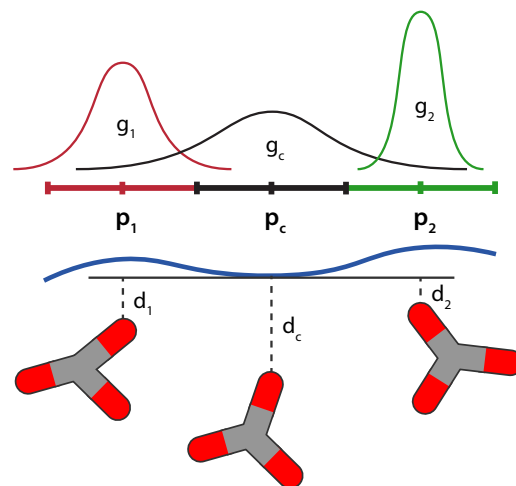
3.3. Transparency and Translucency

In this section, we describe the algorithms developed to visualize the interior structures beneath the molecular surface. With this objective, we support transparencies to visualize interior structures and we simulate translucency effects to only visualize those which are close to the surface.

Our approach is composed of two passes. In the first pass, we render the atoms in a frame buffer storing, per each pixel, the color of the atom and its depth value, d_a. In the second pass, we render our SES using the algorithm described in Section 3.2. Then, for each pixel, we retrieve the values stored by the first pass in the frame buffer and compare the stored depth, d_a, with the depth of the surface, d_s. Using these two values, we compute f_c:

$$f_c = e^{-5\sigma} \qquad \sigma = \max\left(\frac{d_a - d_s}{d_{max}}, 1\right) \qquad (3)$$

where d_{max} is the maximum depth at which we want to visualize the internal structures. f_c is used then to interpolate between the color of the atom and the color of the surface, and thus generate the final color of the pixel. We used an exponential function to compute this factor since it is close to the Beer-Lambert Law [Bee52] that explains the fading of light inside a volume.

The value of d_{max} can be modified in order to modify the transparency effect. Figure 6 presents two images of the same surface with $d_{max} = 6\,\text{Å}$ (left) and $d_{max} = 12\,\text{Å}$ (right). Note how the objects fade out with their distance to the surface. In order to be consistent with the subsurface scattering algorithm, we use $d_{max} = kl_s$, where k is a user-defined constant. This allows us to modify the transparency of the material at the same time as we modify the subsurface scattering effect.

Figure 6: *Our algorithm to visualize internal structures applied to the same area of a molecular surface with different levels of transparency. The level of transparency in the left image is equal to $6\,\text{Å}$ whilst the right image was generated with a transparency level of $12\,\text{Å}$.*
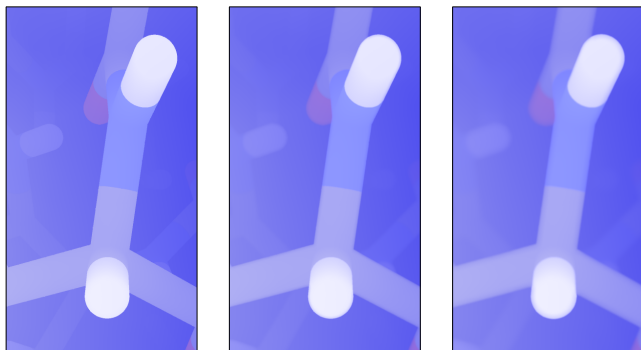
Figure 8: *Results of our blurring algorithm for different kernel sizes. From left to right, kernel size of 3×3 ($n = 1$), kernel size of 11×11 ($n = 5$), and kernel size of 21×21 ($n = 10$). Note how structures buried inside the SES appear more blurred than structures close to the surface.*

Figure 9: *Refractions are calculated when the viewing ray $\vec{v}$ hits the SES volume (grey). Without refractions, we would perform a texture lookup in the background image (blue) containing the structural elements at the same screen coordinates as the pixel is rendered (t). Due to refractions, the viewing ray is bent towards the negative of the surface normal $\vec{n}$ (dashed line). We use Δt to adjust the screen coordinates t to get the pixel value at t'.*

Besides visualizing the interior structures with transparencies, we also simulate the blurring effect of translucent materials. This effect makes the internal structures further inside of such materials appear more blurred than the structures close to the surface. We simulate this effect by the following algorithm: For each pixel p_c, we not only compute its f_c value, but also the f_{c_i} value for all the neighboring pixels, p_i, in a certain region (n neighboring pixels in each direction, i. e. kernel size $(2n+1) \times (2n+1)$). Then, we compute the color c_i of all neighboring pixels by interpolating using their f_{c_i}. Lastly, we determine the final color of pixel p_c by computing a weighted sum of all the colors of neighboring pixels:

$$c = \frac{\sum_{i=1}^{N} c_i * \omega_i}{\sum_{i=1}^{N} \omega_i} \qquad N = (2n+1) \times (2n+1) \qquad (4)$$

The coefficients ω_i control how much a neighboring pixel p_i will contribute to the color of the central pixel p_c. Therefore, these coefficients have to depend on the depth of the pixel p_i and its screen distance to the central pixel p_c. We compute ω_i by using a 2D Gaussian function centered at pixel p_i with the following form:

$$\omega_i = \frac{1}{2\pi\sigma_i^2} e^{-\left(\frac{x^2+y^2}{2\sigma_i^2}\right)} \qquad (5)$$

In this equation, x and y are the normalized distances in the two respective axis between the pixels p_c and p_i. Note that σ_i is the value used to compute f_{c_i} for each pixel p_i as described in Equation 3. Therefore, pixels with high depth values will use a wider Gaussian to compute its factor ω_i, ensuring that the color of a given pixel will be distributed over its neighbors based on its distance to the molecular surface (see Figure 7). Figure 8 illustrates this effect by varying the number of neighboring pixels visited.

We have implemented this algorithm in the same pass in which we render the surface. We trace a ray from the pixel position until we reach the surface. Then we compute the subsurface scattering
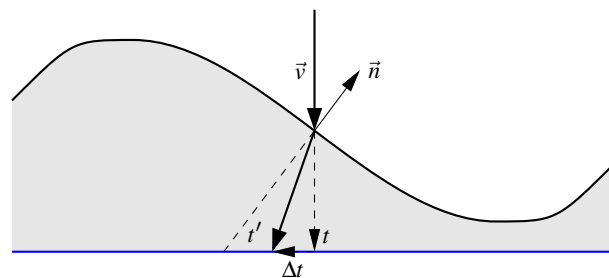
value for the intersecting position, and lastly, we apply the filter described above to visualize the internal structures. Note that this algorithm visits $(2n+1)^2$ neighbors for each pixel, which can be computationally expensive for larger values of n. However, as we show in Figure 15, we still achieve high frame rates for $n = 10$.

3.4. Reflections and Refractions

By using our method to simulate subsurface scattering we introduce information about the thickness of the surface in our visualizations. However, by removing Blinn-Phong from our shading, we also remove information about the curvature of the surface. In order to recover it without introducing misleading information, we incorporated reflections and refractions into our visualization. Reflections simulate the reflection of light by the surface of objects whilst refractions simulate the bending of the light ray when it passes from one medium into another. Since these methods depend on the normal of the surface, they are the perfect candidates for communicating the variation of these normals.

We achieve reflections by computing the reflection vector at the intersecting point **x** on the surface and, then, using sphere tracing again in this new direction as we do with the primary rays. Since our subsurfaces scattering technique only requires taking one more sample, we also calculate this value at the new intersecting point of the reflection vector. The main drawback of this approach is that we do not take into account other structures besides the surface when computing the reflections. If domain experts are visualizing the interaction between the surface and another molecule (as in Figure 1), this molecule will not be reflected by our method.

To achieve refractions we used a screen-space approach similar to existing methods [DW07, Wym05]. At each intersecting point with the surface, we compute a refraction vector based on the normal and the refractive indices of the two mediums. Using this vector, we compute a displacement in screen coordinates (Δt in Figure 9). The pixel at the resulting screen coordinates is then used as the central pixel p_c in the transparency algorithm described in Section 3.3. This algorithm is able to approximate the light refraction effect improving at the same time the perception of the curvature
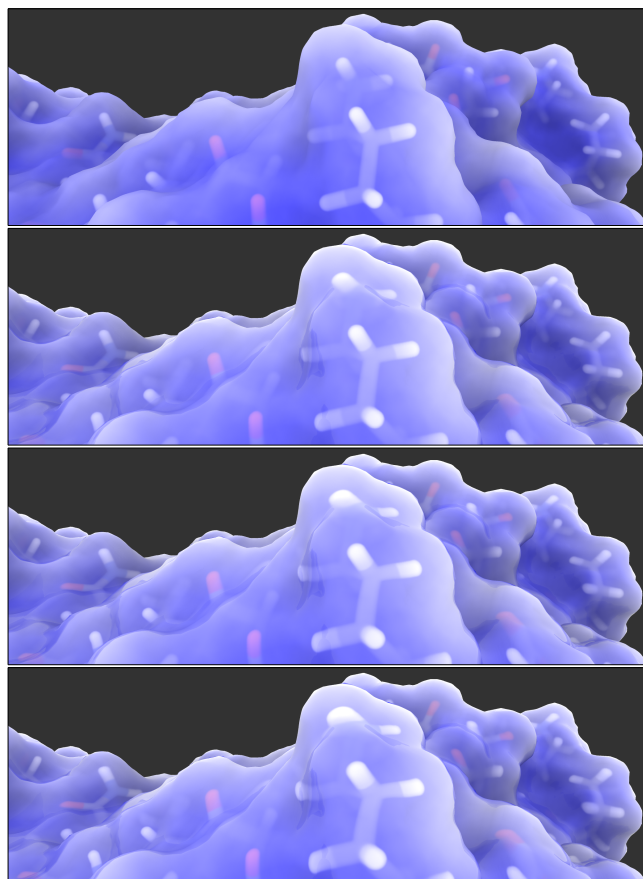
Figure 10: *Reflections and Refractions added to our visualization. The top image presents our subsurface scattering algorithm visualizing at the same time the interior structures. On the other hand, the three other images present the same molecule after applying reflections and refractions with a different index of refraction (*1.15*,* 1.3*, and* 1.5 *from top to bottom). Note how the shape of the molecule is perceived clearly after applying reflections and refractions.*

of the surface. Since we do not perform a real ray-casting, it can produce incorrect results. However, for low refraction indices as the ones we use (the images in this paper were generated using refraction indexes between 1.15 and 1.5), this algorithm generates realistic visualizations at a low computational cost. Note that since the results of our refraction algorithm depend on the screen space displacement used, the refraction index cannot be compared with refraction indices of real materials. Nevertheless, this parameter allows us to control the amount of refraction used in our visualizations. Figure 10 presents the results obtained with different refraction indices.

3.5. User-Guided Filtering

The proposed methods are able to visualize the structures just beneath the molecular surface providing extra information about their chemical properties without cluttering the image with irrelevant interior structures. However, the number of elements on the screen

can still be too much when researchers are interested only in a certain area of the surface. Following the suggestion of several domain experts, we allowed filtering of the internal structures based on the proximity to a given point. In our examples, we used the position of the ligand (drug) during a docking simulation as our point of reference for the filtering. Nevertheless, the same method can be applied to visualize structures near a cavity, a tunnel or an area of interest.

We achieve the filtering by computing the normalized distance between the reference point and the point on the surface in a similar way as we computed σ in Equation 3. Then, as in Equation 3, we determine the fading factor of the pixel by an exponential function. The fading factor is finally used to linearly interpolate between the surface color and the color of the internal structures. Figure 1 presents an example of this algorithm applied to the visualization of a molecular simulation.

4. Evaluation

We evaluated our visualization algorithms at three different levels. First, we obtained feedback from domain experts from two different research groups which helped us to improve our algorithms and determine its possibilities. Second, we carried out two crowdsourced user studies which determined the perceptual improvement introduced by our algorithms. And lastly, we evaluate the performance of these methods by measuring the milliseconds required to generate an image for different configurations. In the next subsections, we describe the results achieved for the different evaluations.

4.1. Expert Feedback

We presented a video with our techniques to five different domain experts in order to gather opinions on the design and implementation of our visualizations. The experts we contacted work in the fields of computer-aided drug design, protein engineering, mutagenesis, and protein unfolding.

All their comments regarding the subsurface scattering were very positive ("I clearly saw the benefits of this new representation", "I think it can be a new technique to add to the existing software"). Some of the experts pointed out that this method significantly increases the shape and topology perception. In particular, they found it very useful to visually identify cavities. One of them, on the contrary, was reticent to use this visualization in his daily research since he argued that existing software already provides enough depth cues, especially when one rotates around the molecule. However, he was very keen on using this visualization to generate publication-quality figures.

Visualizing the atoms beneath the surface was the visualization that they found more useful. All of them agreed that this technique removes a lot of visual noise generated by other techniques. Moreover, they also commented that this technique can help in the analysis of cavities and molecular simulations, especially when domain experts zoom in to analyze a particular region. As a result of this feedback, several suggestions came out, such as the *User-Guided Filtering* (see Section 3.5).

Lastly, we obtained feedback from them regarding reflections and refractions. However, most of them were not able to describe the benefits of this technique.

4.2. User Studies

All domain experts found that our visualization of internal structures improved the current methods based on standard transparency algorithms. Furthermore, most of them found the subsurface scattering algorithm very useful. However, one of them did not find a real perceptual improvement compared to the most commonly used shading techniques available in the molecular visualization packages such as Blinn-Phong. Moreover, none of them were able to identify the benefits introduced by reflections and refractions.

In order to evaluate the perceptual benefits of these techniques, we carried out two crowd-sourced user studies. For this purpose, we used Amazon's Mechanical Turk. The viability of Mechanical Turk as a platform to conduct large-scale perceptual studies has been previously proved by Heer and Bostock [HB10]. This platform allowed us to gather results from users of a broad range of age, gender and nationality [RIS*10]. In the following paragraphs we describe, for the two studies we carried out, their design and the analysis of their results:

Subsurface Scattering Evaluation: First, we evaluated the benefits of our subsurface scattering algorithm when used to visualize molecular surfaces instead of the most commonly used technique, Blinn-Phong. Our hypothesis for this study was that our subsurface scattering algorithm is able to improve the identification of the back shape of the molecular surface in respect to a standard illumination models such as Blinn-Phong. In order to test our hypothesis, we selected seven visual configurations for different molecules. For each configuration, we generated an image using only the Blinn-Phong illumination model and another one using our algorithm of subsurface scattering. Then, we drew a vertical line inside a rectangle as is illustrated in Figure 11.

We showed these 14 different configurations to our users (shuffled using Latin Squares) and we asked them to guess the silhouette of the back part of the surface at the red dotted line. We provided three different options (also shuffled) from where they had to select one. Figure 11 presents one of these configurations and the three available answers. Moreover, we provided an initial training in which we described how these silhouettes were generated. In order to avoid *click-through* users, we introduced two control questions in which we presented one silhouette with the correct answer and two artificial geometric shapes (triangle and rectangle). If a user failed one of these questions, we did not take him into account during the analysis of our results.

We collected data from 75 participants. We recorded for each participant the percentage of right answers with one method and the other. The mean value for the Blinn-Phong model was 39.8% of right answers with a standard deviation of 16.1%. For the Blinn-Phong images there was no direct information about the back shape. Therefore, as we expected, the mean to be similar to randomly selecting one option among three possibilities, $1/3 = 0.333$. On the other hand, the mean obtained with our algorithm was 52.1% of right answers with a standard deviation of 20.1%. Figure 12 presents the plots of both results. Moreover, we performed further tests in order to assure that this difference was significant. First, we tested for normality the differences in the ratio of success between both methods, and we concluded that our data was normally distributed with a 95% confidence interval. Then, since our data was normally distributed, we ran the *paired t-test*. Our hypoth-
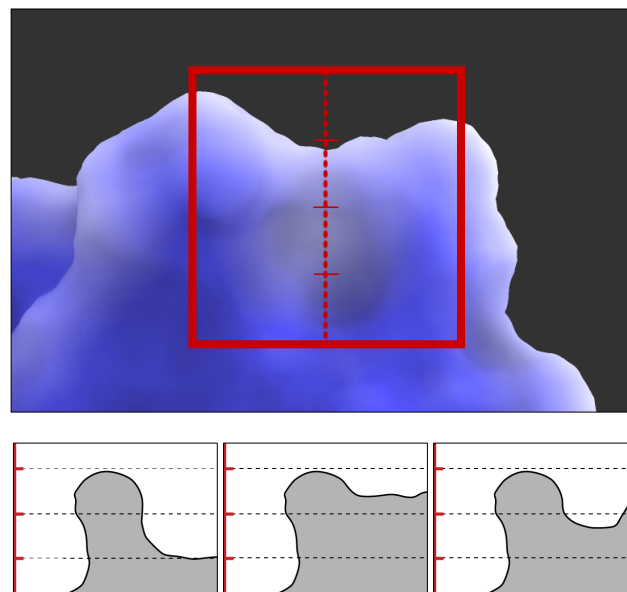

Figure 11: *One of the tests that the participants of our study had to perform. We showed them a big picture of a part of a molecule (top) with a region highlighted in red. The participants then had to guess the contour of the molecule along the dashed red line. They could choose from three possible answers (bottom row) that had all the same contour in the visible part of the image. The part of the contour that was occluded in the image differed in the three choices. In the example presented in this figure, the participants had to guess, based on the white color along the dashed line generated by the subsurface scattering algorithm, that the surface in this area was thinner than the rest of the molecule. Therefore, they should have selected the first option (left).*

esis was that the mean of the difference in the ratio of success was different than 0. Using this test, we were able to determine, based on our collected data, the real mean of this difference. We obtained that the mean for our collected data was 12.19 with a standard deviation of 25.26. Moreover, the test concluded that our difference was statistically significant. The *p*-value obtained for this test was smaller than 0.0001 (at a 0.05 level of significance), which rejects the null hypothesis that the mean of the difference in accuracy is equal to 0. The test determined that the real mean is between 6.38% and 18.01% with a 95% confidence interval, i. e. our visualization method achieves on average an increment in accuracy between 6.38% and 18.01% over Blinn-Phong shading.

As we expected, our method significantly improves the perception of the occluded topology of molecular surfaces. Although other factors can influence the identification of such shapes (such as the surrounding geometry), under the same conditions, our method performed better.

Reflections and Refractions: We also carried out another user study in which we wanted to test the hypothesis that, by adding reflections and refractions to our subsurface scattering visualization, we improved the perception of the surface's curvature.

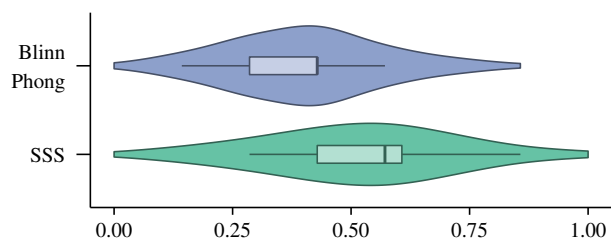Similar to the previous study, we selected seven visual config-

Figure 12: *Violin plots of the results obtained for the user study which evaluated the perceptual improvement of our subsurface scattering algorithm respect to the Blinn-Phong illumination method. As we expected, subsurface scattering significantly improved the identification of the topology of occluded parts of the molecule.*

Figure 14: *Violin plots of the results obtained for the user study in which we evaluated the perceptual improvement of incorporating reflections and refractions to our visualization. As we expected, adding reflections and refractions increased the curvature perception. Moreover, we performed further analysis which proved that this improvement was statistically significant.*

urations of different molecules. For each configuration, we, then, generated two images, one using subsurface scattering with transparencies and another one adding reflections and refractions. As in the previous study (see Figure 11), we marked with a red dotted line the silhouette the users had to determine and we provided three possible answers. Moreover, we introduced two control questions to discard *click-through* users.

We presented the 14 images (shuffled again using Latin Squares) to 72 different users. For each user, we recorded his accuracy with both methods. In order to analyze the obtained results, we followed the same workflow as with the previous study. The mean value of the accuracy for the subsurface + transparencies was 56.8% with a standard deviation of 22.2%. On the other hand, the mean obtained with reflections and refractions was 70.0% with a standard deviation of 23.6%. Figure 14 presents the plots of both groups. As in the previous study, we also performed further tests to assure that this difference was significant. First, we tested if the difference between both methods followed a normal distribution. The result of this test determined that these values followed a normal distribution with a 95% confidence interval. Then, we performed the *paired t-test*. Our hypothesis was that the mean of the difference in accuracy between
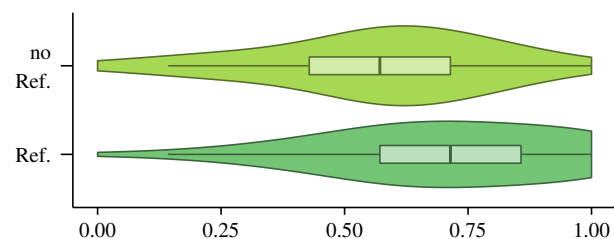
using reflections and refractions and not using them was different than 0. With this test, as before, we can determine the real mean based on our sampled data. The mean of this difference for our collected data was 13.29 with a standard deviation of 28.8. Moreover, the test concluded that our difference was statistically significant. We obtained a *p*-value smaller than 0.0005 (at a 0.05 level of significance), which rejects the null hypothesis that the mean of the difference in accuracy between using refractions and not using them was equal to 0. The test also determined that the real mean is between 6.52% and 20.06% with a 95% confidence interval, i. e. adding reflections and refractions to our visualization achieves on average an increment in accuracy between 6.52% and 20.06% over subsurface scattering and transparencies alone.

Again, as we expected, by adding reflections and refractions we are able to improve the perception of the surface's curvature.

4.3. Performance

In this section, we describe the results obtained for the tests we executed for different molecules. All these tests were carried out in a computer with the following configuration: Intel i7-4790 CPU (3.6 GHz), 16 GB of RAM, Nvidia GeForce GTX 980 with 4 GB of memory and a screen resolution of 1280×720.

Figure 15 presents the results of our tests. We evaluated the performance of our techniques for four molecules of different sizes: 1JJJ (2093 atoms, see Figure 3), 4NKG (4507 atoms), 1AF6 (10517 atoms) and 5XYU (45157 atoms, see Figure 13). For each molecule, we compute the milliseconds required to draw a frame. We rendered the molecule from 512 different random view directions uniformly distributed over a sphere. We measured the milliseconds required to render the molecules from each of these views and compute an average value. In this images, molecules were rendered with a zoom level in which they occupied as much viewport as possible without being clipped.

First, we evaluated the time required to draw a frame using sphere tracing, ambient occlusion and the Blinn-Phong illumination model (bar labeled as 'Blinn-Phong' in Figure 15). We achieved between $3.7\,ms$ and $5.6\,ms$ for the different molecules, which guarantees a frame rate of at least 175 FPS. Second, we evaluated the performance of substituting the Blinn-Phong illumination
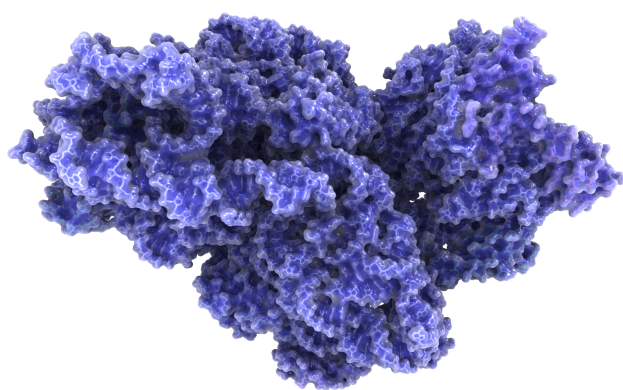

Figure 13: *Molecule 5XYU (45157 atoms) rendered using our techniques: subsurface scattering, transparencies (n = 5), reflections, and refractions.*
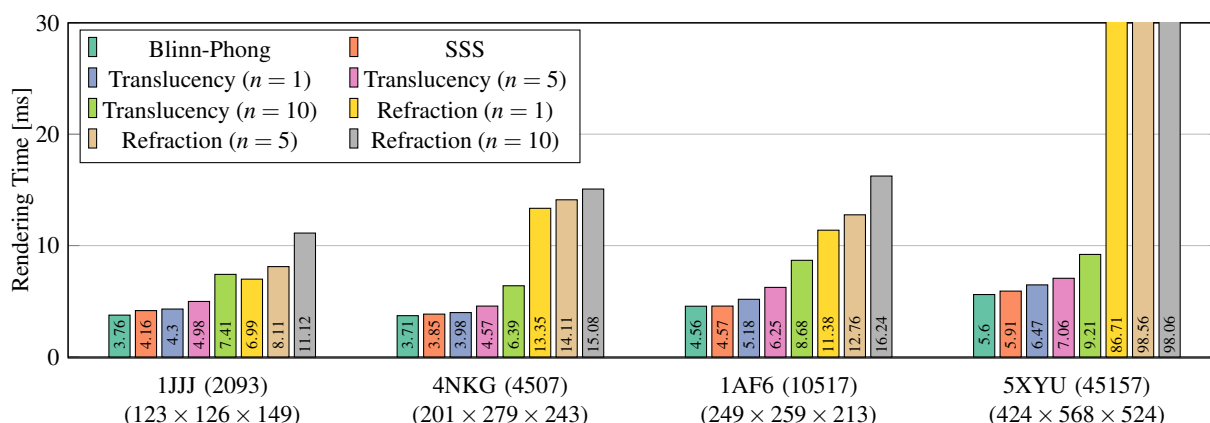
Figure 15: *Average rendering times for four example molecules. The first number in parentheses is the number of atoms of the molecule, the second parentheses have the grid size used to store the SES. We measured timings for simple Blinn-Phong rendering (with ambient occlusion), subsurface scattering (SSS), subsurface scattering and translucency with n samples (Translucency), and all our effects including Refractions and Reflections with n samples (Refraction). All the images were rendered with a resolution of 1280×720 pixels. The large grid size of the last molecule (5XYU) is the reason for the very high rendering times (90ms) using the method with reflections and refractions (Refraction) as those require us to do another ray cast which is slow in large grids.*

model by our subsurface scattering. As we expected, we achieve a similar results since we removed some vector computations of the Blinn-Phong model and only added a texture fetch (information that in most of the cases will reside in the texture cache). For this algorithm, we obtained times between $3.8\,ms$ and $5.9\,ms$. Third, we evaluated the performance of our transparency algorithm. Since the performance of this method is bound by the size of the kernel's filter, we measured the performance of the method with different values of n. We obtained values for $n = 1$ (kernel size of 3×3), $n = 5$ (kernel size of 11×11) and $n = 10$ (kernel size of 21×21). For $n = 1$, adding transparencies almost did not decrease our performance. We measured times between $3.9\,ms$ and $6.4\,ms$. On the contrary, for bigger values of n, we measured an increase of the rendering time between 18% and 37% for an $n = 5$, and 55% and 89% for an $n = 10$ respect to our basic subsurface scattering algorithm. Lastly, we evaluated the performance of the algorithm to simulate reflections and refractions. From our tests, we observed that adding reflections and refractions to our visualization increased the milliseconds between 50% and 62% for molecule 1JJJ. For molecule 4NKG, timings increased between 135% and 235%, and for molecule 1AF6 between 86% and 120%. However, for molecule 5XYU the increase was about 1000%. Whilst we still obtained real-time frame rates for molecules 1JJJ, 4NKG, and 1AF6, for molecule 5XYU, we only achieved around $10\,$FPS. This drop in performance can be explained by the technique we used to approximate reflections. This technique traces secondary rays after hitting the surface which is computationally expensive. For molecules 1JJJ, 4NKG and 1AF6, this computation can be performed in real-time since the resolution of the distance field is between 128^3 and 256^3 grid points. However, for molecule 5XYU, the resolution of the distance field is around 512^3 which requires more texture accesses during ray traversal.

Moreover, since we added extra passes to the SES computation we also evaluated its performance. For a molecular simulation com-

posed of a protein of around 4000 atoms, we required a distance field of a resolution equal to $153 \times 157 \times 203$ in order to store the SES. The computation of the distance field took around $80\,ms$, between $5\,ms$ and $6\,ms$ for the algorithm described by Hermosilla et al. [HKG*17] plus between $77\,ms$ and $78\,ms$ for the 50 extra passes used to propagate the distances along the texture. We chose to perform a number of iterations equal to the maximum number of voxels per axis divided by 4, which provided good results for the molecules tested. Therefore, for such molecular sizes, we are able to compute the distance field without losing interactivity, thus supporting interactive analysis of molecular simulations.

5. Conclusions and Future Work

We have proposed a set of fast methods to compute illumination effects for transparent and translucent materials to improve the visualization of molecular surfaces. First, we proposed an algorithm to simulate subsurface scattering in molecular surfaces, providing information of the thickness of the surface. Moreover, we presented a method to visualize the structures beneath the surface, reducing thus the visual noise usually generated by state of the art methods. We also introduced reflections and refractions to our visualization which improved the perception of the curvature of the surfaces. Although some of the proposed techniques were designed to work on signed distance fields, we described several methods that can be used to apply our techniques to other data types. We evaluated our techniques at three different levels. First, we presented them to domain experts who expressed their interest in our visualizations and their possible applications. Second, we carried out two different crowd-sourced user studies which determined a significant perception improvement by using our visualizations. Lastly, we evaluated their performance by measuring the time required for rendering.

Although our technique to generate reflections and refractions improved the curvature perception in our visualizations, in the fu-

ture, we would like to study the difference of perception quality between this technique and the Blinn-Phong illumination model. Furthermore, we would like to adapt our visualizations to other types SES representations mentioned in the paper (e.g. triangle meshes).

Acknowledgements

References

[Bee52] BEER A.: Determination of the absorption of red light in colored liquids. *Ann. Phys. Chem 86* (1852), 78–88. 5

[Bli82] BLINN J. F.: Light reflection functions for simulation of clouds and dusty surfaces. In *Proceedings of the 9th annual conference on Computer graphics and interactive techniques - SIGGRAPH '82* (New York, New York, USA, 1982), vol. 16, ACM Press, pp. 21–29. 3

[Cha60] CHANDRASEKHAR S.: *Radiative Transfer*. Dover Publications, Inc., New York, 1960. 3

[DI11] D'EON E., IRVING G.: A Quantized-diffusion Model for Rendering Translucent Materials. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 56:1–56:14. 3

[DW07] DAVIS S. T., WYMAN C.: Interactive refractions with total internal reflection. In *Proceedings of Graphics Interface 2007* (New York, NY, USA, 2007), GI '07, ACM, pp. 185–190. 3, 6

[ED08] ERLEBEN K., DOHLMANN H.: Signed distance fields using single-pass gpu scan conversion of tetrahedra. In *GPU Gems 3*, Nguyen H., (Ed.). Addison-Wesley, 2008, pp. 741–763. 3

[Eva06] EVANS A.: Fast approximations for global illumination on dynamic scenes. In *ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 153–171. 3

[HB10] HEER J., BOSTOCK M.: Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 203–212. 8

[HCJ13] HABEL R., CHRISTENSEN P. H., JAROSZ W.: Photon beam diffusion: A hybrid monte carlo method for subsurface scattering. In *Proceedings of the Eurographics Symposium on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2013), EGSR '13, Eurographics Association, pp. 27–37. 3

[HK93] HANRAHAN P., KRUEGER W.: Reflection from layered surfaces due to subsurface scattering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques - SIGGRAPH '93* (New York, New York, USA, 1993), ACM Press, pp. 165–174. 3

[HKG*17] HERMOSILLA P., KRONE M., GUALLAR V., VÁZQUEZ P.-P., VINACUA À., ROPINSKI T.: Interactive gpu-based generation of solvent-excluded surfaces. *The Visual Computer 33*, 6 (2017), 869–881. 2, 3, 4, 10

[HSK89] HART J. C., SANDIN D. J., KAUFFMAN L. H.: Ray tracing deterministic 3-d fractals. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1989), SIGGRAPH '89, ACM, pp. 289–296. 2, 3

[JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. *Siggraph* (2001), 511–518. 3

[JPSK16] JURČÍK A., PARULEK J., SOCHOR J., KOZLIKOVA B.: Accelerated visualization of transparent molecular surfaces in molecular dynamics. In *2016 IEEE Pacific Visualization Symposium (PacificVis)* (April 2016), pp. 112–119. 2, 3

[JZJ*15] JIMENEZ J., ZSOLNAI K., JARABO A., FREUDE C., AUZINGER T., WU X.-C., VON DER PAHLEN J., WIMMER M., GUTIERREZ D.: Separable Subsurface Scattering. *CGF 34*, 6 (Sept. 2015), 188–197. 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. 2, 3

[KDE10] KRONE M., DACHSBACHER C., ERTL T.: Parallel computation and interactive visualization of time-varying solvent excluded surfaces. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology* (New York, NY, USA, 2010), BCB '10, ACM, pp. 402–405. 2

[KKL*16] KRONE M., KOZLÍKOVÁ B., LINDOW N., BAADEN M., BAUM D., PARULEK J., HEGE H.-C., VIOLA I.: Visual analysis of biomolecular cavities: State of the art. *Computer Graphics Forum 35*, 3 (2016), 527–551. 5

[KKP*13] KAUKER D., KRONE M., PANAGIOTIDIS A., REINA G., ERTL T.: Rendering Molecular Surfaces using Order-Independent Transparency. In *Eurographics Symposium on Parallel Graphics and Visualization* (2013), Marton F., Moreland K., (Eds.), The Eurographics Association. 2, 3

[LBH14] LINDOW N., BAUM D., HEGE H.: Ligand excluded surface: A new type of molecular surface. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec 2014), 2486–2495. 2

[LBPH10] LINDOW N., BAUM D., PROHASKA S., HEGE H.-C.: Accelerated visualization of dynamic molecular surfaces. *Computer Graphics Forum 29*, 3 (2010), 943–952. 2, 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

[MM17] MCGUIRE M., MARA M.: Phenomenological transparency. *IEEE Transactions of Visualization and Computer Graphics 23*, 5 (May 2017), 1465–1478. 3

[PGH*04] PETTERSEN E. F., GODDARD T. D., HUANG C. C., COUCH G. S., GREENBLATT D. M., MENG E. C., FERRIN T. E.: Ucsf chimera - a visualization system for exploratory research and analysis. *Journal of Computational Chemistry 25*, 13 (2004), 1605–1612. 3

[PV12] PARULEK J., VIOLA I.: Implicit representation of molecular surfaces. In *2012 IEEE Pacific Visualization Symposium* (Feb 2012), pp. 217–224. 2, 3

[Ric77] RICHARDS F. M.: Areas, Volumes, Packing, and Protein Structure. *Annual Review of Biophysics and Bioengineering 6*, 1 (1977), 151–176. 1

[RIS*10] ROSS J., IRANI L., SILBERMAN M. S., ZALDIVAR A., TOMLINSON B.: Who are the crowdworkers?: Shifting demographics in mechanical turk. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI EA '10, ACM, pp. 2863–2872. 8

[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. 2, 3

[SPG03] SIGG C., PEIKERT R., GROSS M.: Signed distance transform using graphics hardware. In *IEEE Visualization, 2003. VIS 2003.* (Oct 2003), pp. 83–90. 3

[TA96] TOTROV M., ABAGYAN R.: The contour-buildup algorithm to calculate the analytical molecular surface. *J. Struct. Biol. 116*, 1 (1996), 138–143. 2, 3

[VBJ*94] VARSHNEY A., BROOKS F. P., JR., WILLIAM J., WRIGHT W. V.: Linearly scalable computation of smooth molecular surfaces. In *In IEEE Computer Graphics and Applications Vol 14* (1994), p. pp. 2, 3

[Wym05] WYMAN C.: An approximate image-space approach for interactive refraction. *ACM Trans. Graph. 24*, 3 (July 2005), 1050–1053. 3, 6

[YHGT10] YANG J. C., HENSLEY J., GRÃIJN H., THIBIEROZ N.: Real-time concurrent linked list construction on the gpu. *Computer Graphics Forum 29*, 4 (2010), 1297–1304. 3