

# Visibility Equalizer

## Cutaway Visualization of Mesoscopic Biological Models

Submission ID 275

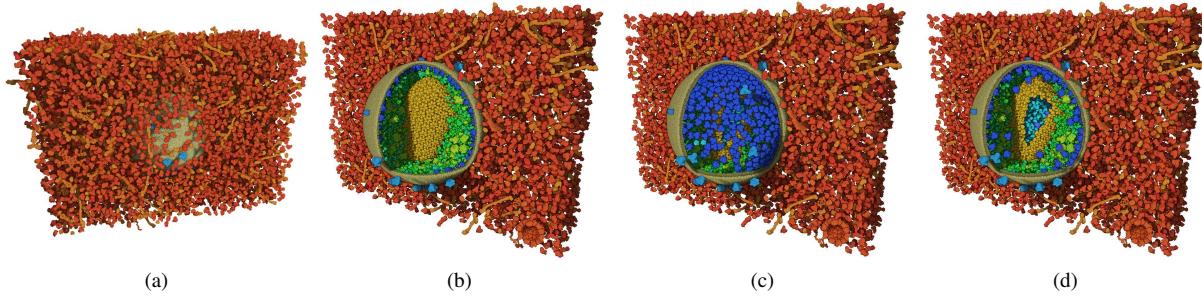


Figure 1: The workflow of our method. (a) Entire dataset is shown. The blood serum (shown in red) is occluding the view of the HIV particle. (b) Clipping objects are shown to selectively cut parts of the data to reveal the HIV capsid. (c) The illustrator decides to show more of the matrix protein (shown in blue), so their clipping is disabled. However, they are now occluding the view of the capsid. (d) The probabilistic clipping has been used to selectively remove those matrix proteins occluding the capsid, but some of them are left in the scene to indicate the presence of this type of protein on the virus membrane. The capsid has been clipped with view-space clipping to reveal its internals.

### Abstract

*In scientific illustration and visualization, cutaway views are often employed as an effective technique for occlusion management in densely packed scenes. We propose a novel method for authoring cutaway illustrations of mesoscopic biological models. In contrast to the existing cutaway algorithms, we take advantage of the specific nature of the biological models. These models consist of thousands of instances that are distributed across a comparably smaller number of different molecular types. Our method constitutes a two stage process. In the first step, clipping objects are placed in the scene, creating a cutaway visualization of the model. During this process, a hierarchical list of stacked bars inform the user about the instance visibility distribution of each individual molecular type in the scene. In the second step, the visibility of each molecular type is fine-tuned through these bars, which at this point act as interactive visibility equalizers. An evaluation of our technique with scientific illustrators confirmed that..... they liked it!!!*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

### 1. Introduction

Molecular biology is an emerging field that is characterized by rapid advances of the current state of knowledge. New discoveries have to be communicated frequently to a large variety of audiences. However, due to their complexity and mesoscopic scale, it is hard to directly convey the discoveries

of molecular phenomena. Therefore illustrations that depict models of these mesoscale structures are the most widely used form of communicating them.

The traditional pipeline for creating scientific illustrations of molecular structures starts with gathering the required knowledge for building models that convey the newly dis-

covered insights. Illustrators then create sketches, in which the relevant internal and external regions of these structures are uncovered. To achieve this, occlusion management techniques, such as *cutaways* are applied. Cutaways remove specific outer parts of the organism model, so that internal structures become visible. When biologists come up with new findings that are not depicted in the existing illustration, the conceptual layout of the original illustration might not be valid anymore. The whole illustration process has to be repeated, in the worst case from the beginning, which is cumbersome and time consuming.

Considering the rapid evolution of knowledge in the field of biology, it is necessary to adapt the traditional illustration pipeline so that new information can be plugged in easily and resulting illustrations can be updated in a very short time. Virtual 3D models of cells and other mesoscale molecular structures can be utilized for these purposes. Biologists have designed tools, such as *cellPACK* [JAAA\*15], to procedurally generate 3D models that represent the structure of micro-organisms such as viruses, or entire cells at atomic resolution. Based on a set of input parameters, individual molecules are assembled into these complex organic static structures. The parameter set consists of a specification of molecular ingredients, concentrations as well as spatial distribution that define where are the instances distributed in a given compartment. The resulting 3D models, in most complex cases, may consist of thousands of ingredients, which in turn, may result in millions of molecule instances and billions of atoms. The instances are densely packed within the predefined compartments, to replicate the actual molecular crowding phenomenon prevailing in living organisms. Due to the high density of these scenes, inner structures that are essential for conveying the functioning of the organism, remain hidden. It is therefore important to develop visualization techniques that would procedurally reproduce the occlusion management methods used in traditional illustration. Currently, this is achieved by placing clipping objects in the scene, which remove specified parts of the displayed model. However, illustrators have to make sure that the essential information, e.g., the ratio of multiple molecular ingredients is represented, and not either hidden in the volume or clipped away (Fig. 2a). To do this, they would need to visually inspect the presence of each single ingredient in the resulting visualization (Fig. 2b).

To alleviate this process, we present our first contribution. We display a stacked bar for each molecular type that encodes the ratio of visible, clipped, and occluded instances of the respective type for the current viewpoint and clipping setting. During the process of placing clipping objects in the scene, these bars therefore continuously reveal, which molecular types are underrepresented or overrepresented. This enables the illustrator to modify the placement of the clipping objects in such a way that every molecular type is adequately represented in the scene. We call this the coarse-level of the visibility specification process.

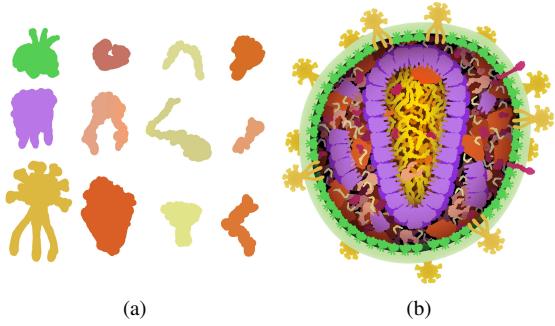


Figure 2: Which of the proteins shown in (a) are visible in the illustration shown in (b) and which are not?

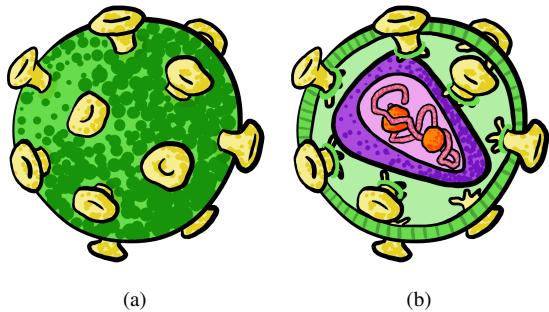


Figure 3: (a) Illustration of an HIV particle. Here, outside membrane of the virus particle is visible. (b) Cutaway view of an HIV particle. Despite the cutaway, some of the glycoproteins (yellow molecules) are kept in the view to provide adequate context.

To preserve important structures that would be removed through clipping objects, such as cutting planes, traditional illustrations also often reintroduce parts of them in front of the revealed cross sections. In Figure 3b, for instance, the glycoproteins (yellow molecules) of the HIV particle which are not occluding the object of interest, in this case the capsid containing the RNA, are left in the illustration to communicate their presence on the surface of the virus (3a). In this way, the main components of the virus particle can be illustrated in a single image. The process of fine-tuning the visibility is extremely time-consuming, as the illustrator has to manually pick individual molecular instances to be reintroduced or removed from the scene.

To significantly speed up this visibility fine-tuning process, we propose the novel metaphor of the *visibility equalizer* as our main technical contribution. To explain its role, we use the metaphor of hi-fi sound reproduction where on a basic level, the *volume control* is used for adjusting the output sound *uniformly* on all frequencies. Such a mechanism corresponds to our coarse-level visibility specifi-

tion through the clipping objects, where all molecular types are uniformly removed from the clipped regions. However, hi-fi sound systems also allow users to fine-tune the sound through an *equalizer*. With an equalizer, the volume of each individual frequency band can be adjusted separately to achieve a desired sound reproduction. In the sense of this metaphor, the visibility of each molecule type can be regarded as a frequency band—represented as the stacked bars that form our visibility equalizer. To achieve a similar level of fine-grained control for the visibility of the molecule types, we make these bars interactive—thus allowing the user to interact with the equalizer. Individual intervals on each bar can be dragged to increase or decrease the visibility of the individual molecular types within the scene, given the specified clipping objects. When interacting with the visibility equalizer, the artist or biologist can intuitively achieve expressive cutaway designs in a fraction of the time that would be needed for manual visibility adjustments.

## 2. Related Work

Related work can be categorized into occlusion management techniques and molecular visualization. We will concentrate on the former, according to the focus of this paper.

### 2.1. Occlusion Management

Related occlusion management techniques can be categorized into object centric approaches and transfer function based approaches. In object centric approaches, the geometry or parts of the volume that are obstructing one or more particular objects of interest are (partially) removed. In transfer function based approaches, the user assigns importances to intervals of the volume data values.

**Object Centered Approaches.** Cutaway and ghosting techniques were first introduced by Feiner & Seligmann [FS92] in 1992. Their work inspired several follow-up approaches [DWE02, DWE03, WEE03, VKG04, VKG05, KTH\*05] that were later summarized in the survey by Viola & Gröller [VG05] under the collective term of *smart visibility* techniques. They coined this term to describe expressive visualization techniques that smartly uncover the most important features of the displayed data, i.e., cutaway views, ghosted views, and exploded views.

Krüger et al. [KSW06] developed a system that applies transparency and shading to enable focus&context visualization in volume data sets with a simple point&click interface. Li et al. [LRA\*07] propose a cutaway design based on the geometry of the occluder in contrast to previous approaches that were based on the occludee. Burns & Finkelstein [BF08] applied the concept of importance-driven cutaways for volume data to polygonal models. Lawonn et al. [LGV\*16] extend this approach to present a composite technique that combines the visualization of blood flow with the surrounding vessel structures. Baer et al. [BGCP11] published a

perceptual evaluation of smart visibility techniques for two ghosted view approaches in comparison to semi-transparent approaches. The results clearly favored the ghosted view techniques. Sigg et al. [SFCP12] propose an approach for automatic cutaway box placement with optimized visibility for target features that are specified as degree-of-interest functions during interactive visual analysis of the volume data. Lidal et al. [LHV12] defined five design principles for cutaway visualization of geological models that we discuss in Section XXX in relation to our approach. The approach by Diaz et al. [DMNV12] preserves the relevant context information in volume clipping by allowing the user to extrude segmented surfaces such as bone structures from the clipping plane.

**Transfer Function Based Approaches.** The context-preserving volume rendering model [BGKG05] uses a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity to selectively reduce the opacity in less important data regions. Contours of surfaces that would be removed due to opacity, remain visible as the amount of illumination received is taken as a measure whether a point should be visible or not. Burns et al. [BHW\*07] propose a multimodal approach that combines CT scan data and real-time ultrasound data. Importance driven shading is used to emphasize features of higher importance that have been revealed through the ghosting.

In his PhD thesis [Vio05], Viola presents an optimization strategy for automatically assigning visual mapping to voxels so that segmented objects in the volume are visible as specified by the user. Correa et al. [CM11] used a similar approach for applying visibility directly to voxels, without the notion of segmented objects. In our approach, we control visibility by interacting with the stacked bars of the visibility equalizer to modify the clipping object properties for each individual molecule type. Ruiz et al. [RBB\*11] propose an approach for automatic transfer function optimization by minimizing the informational divergence between a user specified target distribution and the visibility distribution captured from certain viewpoints.

Transfer function based approaches are well suited for volumetric data that contains segmentable structures, such as the organs or bones in a medical scan. For molecular data this only holds partially true, as some types of molecules do indeed form continuous structures that could be made visible with a transfer function (e.g., membranes, nucleus). On the other side, within these structures there is a more noise-like distribution of these molecules that cannot be segmented into solid structures. In regard to object centered approaches, (partial) occlusion of individual molecules is not an issue as the data does not contain large singular entities such as polygonal or segmented volumetric objects where each single one has a semantic meaning. Instead there are thousands or hundreds of thousands of instances that belong to a couple of dozen molecule types. Our approach is therefore funda-

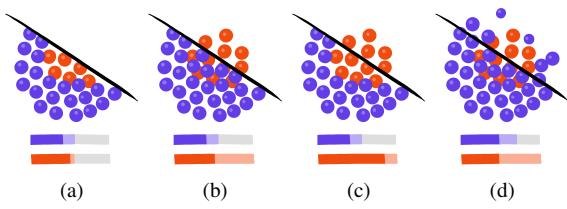


Figure 4: Interaction with the visibility equalizer. (a) increasing the visibility ratio of the red molecules, reduces the occluding blue molecules. (b) completely decreasing the clipping ratio of the red molecules. (c) partially decreasing the clipping ratio of the blue molecules. (d) combining the steps from a and b to increase the overall visibility of the red molecules.

mentally different from existing occlusion management approaches as it combines principles from object centered and transfer function approaches.

## 2.2. Visualization of Molecular Structures

Lindow et al. [LBH12] were the first to introduce a fast method for the real-time rendering of large-scale atomic data on consumer level hardware. They utilize instancing on the GPU to repeat these structures in the scene. For each molecule type, a 3D grid of the atoms is created and stored on the GPU. Falk et al. [FKE13] further refined the method with improved depth culling and hierarchical ray casting to achieve faster rendering performance for even larger scenes.

Other related work is concerned with illustrative molecular visualization. Grottel et al. [GKSE12] and Eichelbaum et al. [ESH13] propose ambient occlusion approaches for large molecular scenes in order to improve the depth perception in these complex structures. Parulek et al. [PJR\*14] propose a continuous level of detail scheme for molecular data that offers gradual shape simplification for distant molecules based on a clustering of the atomic spheres.

In the domain of dedicated mesoscopic molecular visualization, our approach is the first to introduce illustrative occlusion management techniques.

## 3. Overview

The two main components of our method are the *clipping objects* and the *visibility equalizer*. The visual encoding of the visibility equalizer is illustrated in Figure 5. It contains a series of stacked bars, one for each molecular ingredient. The stacked bars show the relation of three visibility values to each other: *a* - the ratio of visible instances of the given ingredient from the current viewpoint; *b* - the ratio of occluded instances; *c* - the ratio of instances which are removed by the clipping objects. The instances are never partially clipped - they are either removed or preserved in the

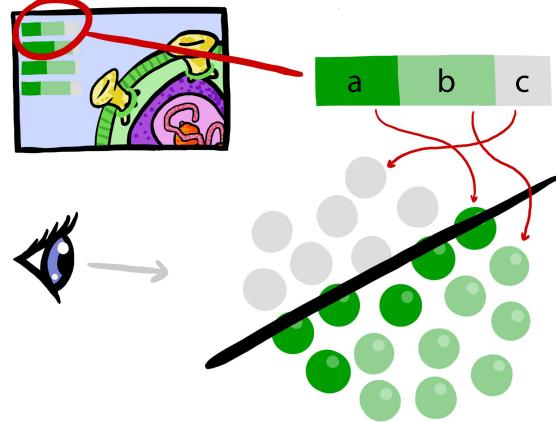


Figure 5: Visual representation of the visibility equalizers. Each molecular ingredient has its own stacked bar showing (a) instances visible from the current viewpoint, (b) occluded instances, (c) instances clipped away by the clipping objects.

scene. This means that we can derive the visibility values by counting the visible and clipped-away instances.

The clipping objects are implicitly represented by signed distance functions. The zero level set represents the surface clipping object. Positive distances lie outside of the object and negative distances within the object. Clipping can occur relative to the object, which we refer to as *object-space clipping* but also relative to the viewpoint, which we describe as *view-space clipping*. For the latter, the shape of the clipping object acts as a focus region that is revealed through clipping based on its projection onto the screen.

Figure 6 depicts the workflow of our method as a state-machine. There are three possible states, denoted as *S1*, *S2*, and *S3*. In the state *S1*, the scene is displayed without any clipping. The visibility equalizer shows that no molecules are clipped, and provides the described visibility information for each molecular ingredient. Clipping objects can be used to transition to state *S2*, where parts of the model are removed in a deterministic way. Here, the distance function of the clipping objects determines whether an instance is clipped or not. The ratios of the clipped instances are displayed by the visibility equalizer. The process of placing and manipulating clipping objects corresponds with the coarse visibility specification as introduced in Section 1.

At any point, the user can interact with the visibility equalizer to adjust the amount of the visible instances in the scene. By dragging the dark green bars, the visibility relative to occluders can be increased or decreased. Dragging the light green bars, modifies the visibility relative to clipping object. When the bars are dragged, probability thresholds of selected clipping objects are modified - directly influencing the probability that an instance of the respective type is

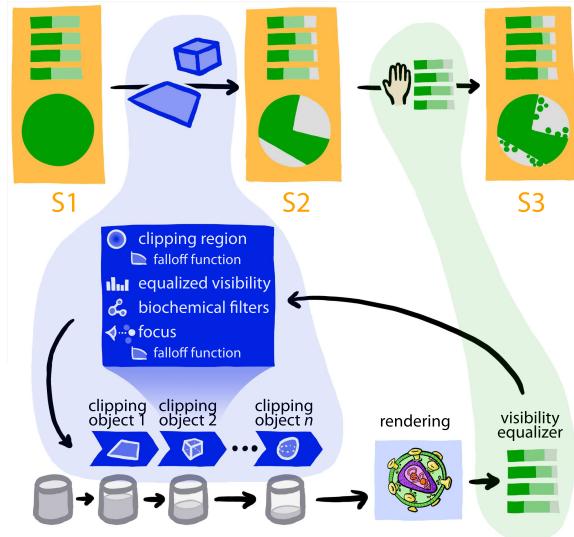


Figure 6: The workflow of our method. The data can be displayed either without any clipping (*S1*), with deterministic clipping defined by the clipping objects (*S2*), or probabilistic clipping specified through the visibility equalizer (*S3*). Lower part (technical overview): each clipping object filters the output of the previous one. The rendered image is used to update the visibility equalizer. Interaction with the bars updates the scene to match the specification in the visibility equalizer.

clipped or not. This complements the deterministic clipping introduced in the coarse step. This situation is represented by the state *S3* and corresponds to the visibility fine-tuning step. The possible interactions with the visibility equalizer are illustrated in Figure 4.

The lower part of Figure 6 shows the technical overview of our method. At the beginning of the pipeline, multiple clipping objects are placed into the 3D scene, filtering the data. Each clipping object filters the output of the previous one. In the rendering stage, the visible and clipped-away instances are counted in respect to the current viewpoint and the visibility equalizer is updated each frame. In case the user interacts with the stacked bars of the visibility equalizer, the probability thresholds of the selected subset of the clipping object are modified. Then the 3D scene is updated to match with the specification in the visibility equalizer.

#### 4. Object-Space Clipping

Clipping objects define which instances of a given ingredient type are displayed. There are two ways of how a clipping object can influence the clipping: either in object-space or in view-space. In this Section, we will explain in detail how clipping objects operate in object-space. Using an object-

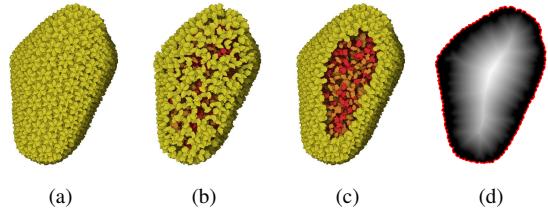


Figure 7: View-space clipping, (a) shows the full HIV Capsid, (b) shows the uniformly distributed clipping, (c) demonstrate the aperture effect and (d) shows the results of the 2D distance transform of the clipping mask.

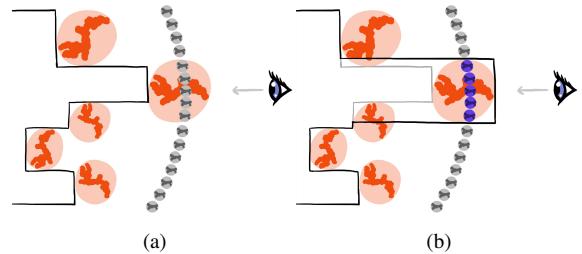


Figure 8: The occlusion clipping principle using a depth-stencil mask, the instances in red constitute the object of focus and the instances in grey are the potential occluders. The bar chart represents the depth-stencil of the mask. In (a) the occluders are overlapping the depth-stencil mask and will therefore be discarded. In figure (b) the instances (in blue) will be preserved after applying a depth bias on the overlapping red instance, allowing us to perform contextual anchoring described in Section 5.3.

space approach, the clipping objects will discard instances independent of the viewing direction.

#### 4.1. Clipping Region Localization

Clipping objects can be associated with geometrical shapes to specify a sub-region of the domain which is influenced by the clipping. Our system currently supports the following set of primitive shapes: plane, cube, sphere, cylinder and cone. The first task of the object-space clipping is to determine whether an instance of a molecule is located in the sub-region defined by the clipping object geometry. To determine if an instance lies inside or outside the clipping object region, we compute the signed distance between the bounding sphere of the instance and the closest point on the surface of the clipping region. To accelerate the computation, we solve the problem analytically using a mathematical description of the 3D signed distance field (SDF). It is also possible to apply simple SDF operators to the distance field, such as translation, rotation and scaling. The clipping region

can also be reversed by inverting the result of the signed distance function, offering users flexibility. For instance, using a spherical shape, the clip region would be set to the inside of the sphere by default, while in inverted mode it would correspond to the outside of the sphere. Although the set of offered clipping shapes is yet limited, it would be trivial to enrich it using more complex SDF operators, such as union, to merge several shapes together in one single distance field, and thus obtain more complex clipping region shapes.

#### 4.2. Clipping Parameters

A clipping object comprises two basic parameters, defined for each ingredient, that control the visibility of instances, based on their type. The first parameter controls the ratio of clipped elements of a given type that are located in the clipping region. We refer to this value as object-space clipping probability. This parameter allows the user to control the degree of *fuzziness* of the clipping, as explained in Figure 4. The other filtering parameters are related to biochemical properties and allow users to control the clipping based on the molecular mass or concentration of given ingredient types. These parameters can be interactively changed via the user interface.

First, filtering is applied based on the clipping probability. For each instance, we compare a uniformly distributed random number with the clipping probability of the instance ingredient type. If the random number is higher than the probability, the instance is marked as discarded, and will not be rendered. The random number is initially set for each individual instance and remains unchanged, in order to avoid getting different results each time. Secondly, instances are filtered according to their biochemical properties, for each cut object and each ingredient type. The user defines a range of values for both quantities and the molecular weight. Instances whose properties lie outside these ranges are marked as discarded and will not be rendered.

#### 4.3. Falloff Function

To further increase the control over the probabilistic clipping, we introduce falloff functions. Falloff functions can be used to modulate the effect of the probabilistic clipping with respect to the distance from the clipping surface. This is easily achievable, since we use distance functions to represent the clipping objects. Therefore, the distance of a molecular instance from the clipping surface can be evaluated by simply sampling the distance function of the given clipping object at the 3D position of the instance. The farther away from the clipping surface the instance is, the lower will be the effect of the probabilistic clipping specified through the visibility equalizer.

The object-space clipping probability of a molecule on the 3D position  $p$  is multiplied by the falloff function  $f(p)$ . The falloff function is defined as follows:

$$f(p) = 1 - \min(1, (d(p)/d)^c) \quad (1)$$

where  $d(p)$  is the distance to the clipping surface from the point  $p$ . The function is parametrized by parameters  $d$  and  $c$ , where  $d$  is the maximum distance up to which the object-space clipping probability takes effect, while  $c$  specifies the exponent of the falloff function.

An example of the falloff function applied to the object space clipping is shown in Figure 9d. The molecules of the blood serum (shown in red) are gradually removed from bottom to top. In this way, the information about the concentration of these molecules is illustrated at the bottom of the scene, while the visibility of the HIV particle (blue and green) is increased towards to top of the scene.

### 5. View-Space Clipping

While object-space clipping using primitive shapes allows for a great degree of flexibility, it requires cumbersome manual operations for complex set-ups, and is also limited in terms of shape diversity. We therefore provide an additional functionality to manually specify a set of ingredient as object of focus, and to selectively remove occluding instances to ensure a maximum degree of visibility. We use occlusion queries to determine what are the instances located in front of the focus region. We provide users the option to control the degree of fuzziness of the clipping, similarly to object-space clipping. We also introduce a new effect inspired from the work of scientific illustrators, that allows users to control the degree of aperture of the view-space clipping. Finally, we tweak our occlusion query method to improve depth perception by performing contextual anchoring of instances that should normally be clipped, when located in the close vicinity of the focus region.

#### 5.1. Occlusion Queries

We perform occlusion queries using an image-based approach on the GPU. We priorly render an off-screen texture containing all the ingredients of the focus object, which we use as a stencil-depth mask to perform occlusion queries. We only render the bounding spheres of the instances, in order to lower the cost of an additional render pass. There can be several ingredient types constituting the object of focus, however, only one mask texture can be generated per clipping object. To approach zero GPU driver overhead, we perform custom occlusion queries in a single draw call, using the programmable graphics pipeline. Subsequently, we draw the bounding sphere of all remaining instances over the mask. Due to early depth-stencil test, implicitly performed by the graphics hardware, fragments of the bounding spheres that will pass the test are therefore guaranteed to be overlapping fragments. We then update the clipping state of the occluding instance by updating a flag stored in the main video

memory directly from the fragment shader. An example of how we perform occlusion queries is shown in Figure XX a.

## 5.2. Clip Parameters

We provide an additional parameter to control the degree of fuzziness of the view-dependant clipping, which we dub view-dependant clipping probability. This value is priorly set by the user for each ingredient type. Once we computed the occlusion state of an instance, we evaluate the clip probability, similarly to the way we evaluate object-space clipping. We compare the clipping probability with an uniformly distributed random number, initially set for each instance to determine if an occluding instance shall remain visible. This will however result in a uniformly distributed number of visible occluders over the object of focus, which might not always be the best design choice because it fragments heavily the overall structure of the occluders compartment, see Figure XX.

We propose an alternative technique for fuzzy removal of occluding instances. We define an additional parameter, called aperture coefficient, which controls the distance threshold in 2D from the centre to the edges of the mask, below which occluding instances shall be clipped. A visual explanation of the aperture effect is shown in Figure X. To enable this effect with compute the 2D distance transform of the occlusion mask which we store in a separate off-line texture. We use the GPU Jump Flooding Algorithm by Rong & Tan [?] to interactively recompute the 2D distance field every frame. After computation of the distance transform, the texture stores for each pixel, the distance to the contours of the shape. Then, while computing occlusion queries in the fragment shader, we may simply look-up the mask contour distance for the current fragment, and discard instances accordingly to the user-defined aperture coefficient, priorly set of each ingredient type.

## 5.3. Contextual Anchoring

When observing still renders of cut-away scenes, it might still be challenging to perceive the depth of objects correctly, despite using lighting-based depth cues. We propose an additional method for depth guidance, which modifies the results of the clipping to preserve elements, located in a close range to non-clipped elements, and that would normally be clipped. This way, the viewer, assuming that he is aware of what has been clipped, will intuitively understand where instances are located due to contextual anchoring. This principle is shown in Figure XX, where we can observe parts of the green membrane anchored around channel molecules, and which indicate that the channel molecules are located on the surface of the object. We were able to procedurally reproduce this effect by applying a depth bias for selected focused molecules to ensure that they will overlap ingredients that need to be preserved. A explanation of the depth

guidance effect is shown in Figure XX and a resulting render in show in Figure XX.

## 6. Equalizing Visibility

To provide a clear overview of the scene properties, we display stack bars for each ingredient type that indicate information about their degree of visibility. We chose to show three ranges for each histogram, respectively corresponding to the number of visible and clipped instances. In order to fill the stack bars with correct values, we perform book-keeping of both clipped and visible instances, which we recompute every frame. The principle of the book keeping is show in Figure ??; The stack bars are also interactive; the user can manipulate the clipping properties of the corresponding ingredient type by dragging the range handles, thus increasing or decreasing the number of displayed elements.

### 6.1. Book-Keeping

Modification of the visibility using the equalizer requires to know for each ingredient type, how many instances have been culled and how many instances are actually visible on the screen, see Figure xx. It is worth mentioning that our system leverages the power of the GPU to compute the clipping state of each instance every frame. Therefore, the information about clipping states is stored in the GPU memory. In order to avoid overhead due to data transfer between CPU and GPU, we perform book-keeping on the GPU using atomic operations. We priorly declare a buffer on the GPU to store the number of clipped and visible instances for each ingredient type. To obtain the number of clipped instances, we simply increment the corresponding counter, each time an instance has been discarded using an atomic addition function.

For computing the number of visible instances, we first need information about actual visibility of rendered instances. We render an additional off-screen texture where each pixel contains the internal ID of the instances. We also declare an additional buffer to store a flag for each instance, which indicates if an instance is visible or hidden. Then, by browsing through each pixel of the aforementioned ID texture in an additional pass, we update the visibility flag for the ID contained in each pixel. Finally, we browse through each instance, and increment the corresponding counter accordingly to the visibility flag using an atomic addition function, similarly to the counting of clipped instances.

### 6.2. User Interaction

Upon manipulation of histogram ranges the system will either increase or decrease the number of clipped instances that correspond to the histogram ingredient type. This is intended to optimise the way user interact with the system, by offering the user to directly manipulating quantities rather

than abstract internal values such as the clip probability. Additionally the user may also manipulate more advanced parameters in an additional UI panel. We chose to map the motion of the histogram handles the clip probabilities. The first handle manipulates the view-space clip probability, while the second handle controls the object-space one. It is worth mentioning that the clip-probabilities that are manipulated by the user correspond to the currently selected clip-object. The histogram view, however, remains the same when a different clip-object is selected.

Quantities are relative by default, i.e., they represent a percentage of the total number of instances for a given ingredient. However, thanks to our design choices, they can also be displayed as absolute quantities with limited additional effort. For displaying absolute quantities we support logarithmic scaling to ensure ingredients present in low quantities to be visible in the histograms. An logarithmic ruler is also provided to help the user understanding the displayed values.

It is also worth mentioning that histograms are displayed in a tree layout where nodes correspond to compartments and leaves to ingredients. Additional histograms are therefore displayed for each compartment by averaging the values of the ingredients contained inside. Upon manipulation of compartment histograms properties, all the children ingredients will be updated accordingly.

## 7. Results and Discussion

We applied our method to three different datasets to showcase the capabilities of our method. The first dataset is a model of an HIV particle in blood serum. In Figure 9 we show an example of a single clipping plane used to reduce the concentration of the blood serum molecules, so that the HIV particle is visible. However, to avoid misleading the viewer about the actual concentration of the blood molecules, we apply ghosting effect. It illustrates what the actual concentration is, while the visual clutter caused by the blood serum is significantly reduced. Figure 9 illustrates that this can be done in different ways according to the artistic vision.

The second dataset to which we applied our method is a model of *Mycoplasma mycoides*. Figure 10b show how clipping object have been used to reduce visual clutter to illustrate the positions of the ribosomes (shown in blue) within the cell.

## 8. Evaluation

Our technique has potential for assisting scientific exploration of large molecular scenes, but it also has potential for educational purposes. Therefore we evaluate the significance of our tool by collecting informal user feedback from domain experts from this two areas. In both cases we did a

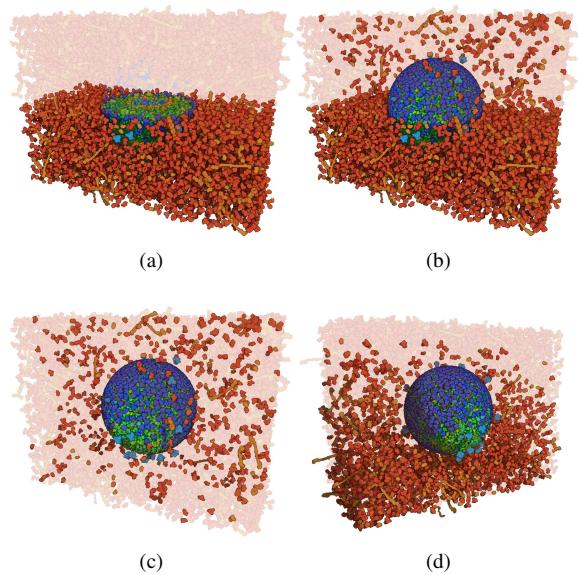


Figure 9: The cutaway molecules are indicated through the ghosting effect. (a) Part of the scene is removed by a clipping plane. (b) The amount of cutaway molecules has been decreased through the probabilistic clipping, which caused some of the cutaway molecules to be reintroduced into the scene. (c) The blood serum (red molecules) are removed by probabilistic clipping in the entire dataset. (d) A falloff function is used to gradually change the influence of the probabilistic clipping.

remote walk-through introduction of our software, while collecting their first impression. Additionally, we gave them an executable version and asked them to provide a short written description of their impression after trying it by themselves. We first sent an early version of our tool to a biomedical illustrator with a strong academic background in chemistry. He has 10 years of experience in the field of computer graphics and he also teaches online paid animation courses at CgSociety focusing on training his students to create accurate representation in biomolecular fields. He also taught course about computer graphics in his own institute (National research council of Italy) focusing on teaching basic skill for lighting and rendering to non computer graphic literate users. Here is a quote from his written feedback:

*“...in my opinion it can be a very useful toolkit for an illustrator in the biomedical field...It also seems very promising for interactive teaching and also for animation purposes... One very useful feature of the software is the possibility to “cut” plane of interest of a particular space keeping the information of all “layers” creating a “gradient” of concentration of the ingredient of the recipe shown. A visualization that reminds an “exploded model” but for biological assem-*

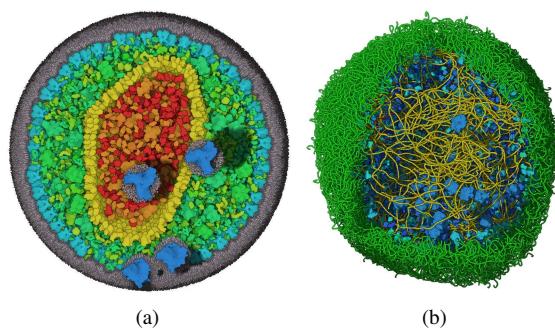


Figure 10: (a) HIV clipped with a plane. Envelope proteins (blue) are reintroduced in the scene. Contextual anchoring is used to illustrate the interaction of the envelope proteins with the membrane. (b) Mycoplasma mycoides.

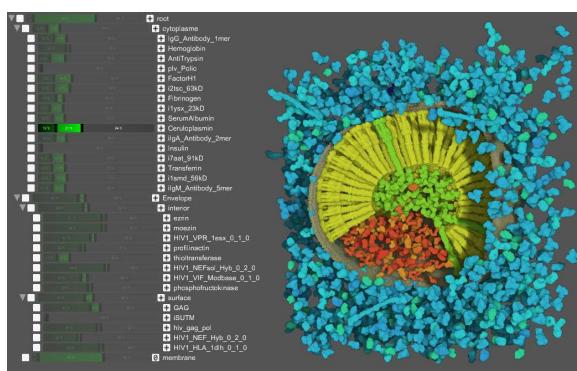


Figure 11: Mature HIV.

*bly and it can be achieved without selecting by hand every instance you would like to hide.”*

The overall feedback from the biomedical illustrator was very positive. He was impressed by how fast and responsive our software was, at manipulating such a great number of instances, in comparison with the animation and rendering tools he normally uses. He also enjoyed the novel concept of fuzziness and gradient clipping, which would be cumbersome to achieve with the tools he currently uses. As a negative point, he deplored the lack of exporting capabilities to mainstream animation packages such as Maya or Cinema 4D.

Secondly, we interviewed an expert in the domain of molecular biology. His research focuses on development and application of computational technologies to study of the structure and function of biological molecules. His area of interest ranges from design of therapeutic agents for AIDS, cancer and heart disease, to development of novel algorithmic, visualization and human interface approaches for molecular biology. In nearly thirty years of experience, he has also contributed to a great number of molecular anima-

tions and visualizations that have appeared in popular books, magazines, newspapers and on television. Here is a quote from the written feedback we collected:

*“...The aperture cutting feature is especially useful for exploring a feature or object in the context of a crowded molecular environment. The ability to shade the model with the aperture cut heightens the impression that the object being viewed is embedded in the volume. Without the shading the object in the aperture appears to be floating above the surrounding environment, when the model is interactively manipulated. The ability to retain a subset of the clipped object (“fuzzy clipping”) is an interesting feature that could be very useful under certain circumstances. If one wants to get an impression of reducing the concentration of some of the molecular ingredients, or of what a gradient of certain molecular ingredients would look like, this feature could be useful.”*

For the second interview, the overall impression of the expert in molecular biology was also quite positive. He also greatly enjoyed how easy and fast it was to perform clipping, and also enjoyed the polished user interface for manipulating the cut object parameters. He also expressed the wish for several additional features to improve how the tool could be employed more efficiently by his peers. The first wish was to add the possibility to filter molecules according their biomolecular properties, such as molecular weight or concentration. He also wished to see the ghosts of clipped instances in order to visually convey the proportions of removed elements in the 3D scene while preserving the current visibility setup. Additionally, he highly recommended us to add directional lighting to the scene in order to improve depth perception. Accordingly to the expert's comments we added these last features in the current version of the software.

To summarize, the evaluation showed that, our equalizer-based approach for visibility specification was valuable and effective for both, scientific and educational purposes according to the opinion of domain experts.

## 9. Conclusions

In this paper, we present a novel method for authoring cutaway illustrations made from mesoscopic biological 3D models. To monitor and fine-tune the process, we introduce the visibility equalizer. It keeps the user informed about the ratios of the molecular instances removed by the clipping objects, or occluded from the current viewpoint. Moreover, the visibility equalizer allows the users to directly change these ratios by interacting with the visibility equalizer, thus fine-tuning the visibility of the individual molecular ingredients within the visualized scene.

The proposed method allows the illustrators to create high quality illustrations from the pre-packed biological models

in a reasonable amount of time. This was confirmed by gathering feedback from the professional illustrators, as well as by recreating existing illustrations with our method. While being simple to use, the method gives the user a considerable level of artistic freedom.

The conceptual contribution of this work is demonstration of a scenario where a visualization metaphor, such as the stacked bar chart in our case, can serve as a user interface for performing a specific task, in our case authoring a cutaway design. We expect to see further cases where the (information) visualization acts simultaneously as the user interface.

## References

- [BF08] BURNS M., FINKELSTEIN A.: Adaptive cutaways for comprehensible rendering of polygonal scenes. In *ACM SIGGRAPH Asia 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH Asia '08, ACM, pp. 154:1–154:7. 3
- [BGCP11] BAER A., GASTEIGER R., CUNNINGHAM D., PREIM B.: Perceptual evaluation of ghosted view techniques for the exploration of vascular structures and embedded flow. *Computer Graphics Forum* 30, 3 (2011), 811–820. 3
- [BGKG05] BRUCKNER S., GRIMM S., KANITSAR A., GRÖLLER M. E.: Illustrative context-preserving volume rendering. In *Proceedings of the Seventh Joint Eurographics / IEEE VGTC Conference on Visualization* (Aire-la-Ville, Switzerland, Switzerland, 2005), EUROVIS'05, Eurographics Association, pp. 69–76. 3
- [BHW\*07] BURNS M., HAIDACHER M., WEIN W., VIOLA I., GRÖLLER M. E.: Feature emphasis and contextual cutaways for multimodal medical visualization. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC Conference on Visualization* (2007), EUROVIS'07, Eurographics Association, pp. 275–282. 3
- [BV09] BIRKELAND A., VIOLA I.: View-dependent peel-away visualization for volumetric data. In *Proceedings of the 25th Spring Conference on Computer Graphics* (New York, NY, USA, 2009), SCCG '09, ACM, pp. 121–128.
- [CM11] CORREA C., MA K.-L.: Visibility histograms and visibility-driven transfer functions. *Visualization and Computer Graphics, IEEE Transactions on* 17, 2 (Feb 2011), 192–204. 3
- [DMNV12] DÍAZ J., MONCLÚS E., NAVAZO I., VÁZQUEZ P.: Adaptive cross-sections of anatomical models. *Computer Graphics Forum* 31, 7 (2012), 2155–2164. 3
- [DWE02] DIEPSTRATEN J., WEISKOPF D., ERTL T.: Transparency in interactive technical illustrations. *Computer Graphics Forum* 21, 3 (2002), 317–325. 3
- [DWE03] DIEPSTRATEN J., WEISKOPF D., ERTL T.: Interactive cutaway illustrations. *Computer Graphics Forum* 22, 3 (2003), 523–532. 3
- [ESELH13] EICHELBAUM S., SCHEUERMANN G., HLAWITSCHKA M.: PointAO - Improved Ambient Occlusion for Point-based Visualization. In *EuroVis - Short Papers* (2013), Hlawitschka M., Weinkauf T., (Eds.), The Eurographics Association. 4
- [FKE13] FALK M., KRONE M., ERTL T.: Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. *Computer Graphics Forum* 32, 8 (2013), 195–206. 4
- [FS92] FEINER S., SELIGMANN D.: Cutaways and ghosting: satisfying visibility constraints in dynamic 3d illustrations. *The Visual Computer* 8, 5-6 (1992), 292–302. 3
- [GKSE12] GROTTEL S., KRONE M., SCHARNOWSKI K., ERTL T.: Object-space ambient occlusion for molecular dynamics. In *Visualization Symposium (PacificVis), 2012 IEEE Pacific* (Feb 2012), pp. 209–216. 4
- [JAAA\*15] JOHNSON G. T., AUTIN L., AL-ALUSI M., GOODSELL D. S., SANNER M. F., OLSON A. J.: cellPACK: a virtual mesoscope to model and visualize structural systems biology. *Nature methods* 12, 1 (Jan. 2015), 85–91. 2
- [KSW06] KRÜGER J., SCHNEIDER J., WESTERMANN R.: Clearview: An interactive context preserving hotspot visualization technique. *Visualization and Computer Graphics, IEEE Transactions on* 12, 5 (Sept 2006), 941–948. 3
- [KTH\*05] KRÜGER A., TIETJEN C., HINTZE J., PREIM B., HERTEL I., STRAUSSG.: Interactive visualization for neck-dissection planning. In *Proceedings of the Seventh Joint Eurographics / IEEE VGTC Conference on Visualization* (Aire-la-Ville, Switzerland, Switzerland, 2005), EUROVIS'05, Eurographics Association, pp. 295–302. 3
- [LBH12] LINDOW N., BAUM D., HEGE H.-C.: Interactive rendering of materials and biological structures on atomic and nanoscopic scale. *Computer Graphics Forum* 31, 3pt4 (2012), 1325–1334. 4
- [LGV\*16] LAWONN K., GLASSER S., VILANOVA A., PREIM B., ISENBERG T.: Occlusion-free blood flow animation with wall thickness visualization. *Visualization and Computer Graphics, IEEE Transactions on* 22, 1 (Jan 2016), 728–737. 3
- [LHV12] LIDAL E. M., HAUSER H., VIOLA I.: Design principles for cutaway visualization of geological models. In *Proceedings of Spring Conference on Computer Graphics (SCCG 2012)* (May 2012), pp. 53–60. 3
- [LRA\*07] LI W., RITTER L., AGRAWALA M., CURLESS B., SALESIN D.: Interactive cutaway illustrations of complex 3d models. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. 3
- [PJR\*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 (Dec 2014), 276–287. 4
- [RBB\*11] RUIZ M., BARDERA A., BOADA I., VIOLA I., FEIXAS M., SBERT M.: Automatic transfer functions based on informational divergence. *Visualization and Computer Graphics, IEEE Transactions on* 17, 12 (Dec 2011), 1932–1941. 3
- [SFCP12] SIGG S., FUCHS R., CARNEKY R., PEIKERT R.: Intelligent cutaway illustrations. In *Visualization Symposium (PacificVis), 2012 IEEE Pacific* (Feb 2012), pp. 185–192. 3
- [VG05] VIOLA I., GRÖLLER M. E.: Smart Visibility in Visualization. In *Computational Aesthetics in Graphics, Visualization and Imaging* (2005), Neumann L., Sbert M., Gooch B., Purgathofer W., (Eds.), The Eurographics Association. 3
- [Vio05] VIOLA I.: *Importance-Driven Expressive Visualization*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, June 2005. 3
- [VKG04] VIOLA I., KANITSAR A., GRÖLLER M. E.: Importance-driven volume rendering. In *Proceedings of the Conference on Visualization '04* (Washington, DC, USA, 2004), VIS '04, IEEE Computer Society, pp. 139–146. 3
- [VKG05] VIOLA I., KANITSAR A., GRÖLLER M. E.: Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics* 11, 4 (July 2005), 408–418. 3

- [WEE03] WEISKOPF D., ENGEL K., ERTL T.: Interactive clipping techniques for texture-based volume visualization and volume shading. *Visualization and Computer Graphics, IEEE Transactions on* 9, 3 (July 2003), 298–312. [3](#)