Eurographics Conference on Visualization (EuroVis) 2016
K-L. Ma, G. Santucci, and J. J. van Wijk
(Guest Editors)

Volume 35 (2016), Number 3

—— Visibility Equalizer ——
Cutaway Visualization of Mesoscopic Biological Models

M. Le Muzic[†1], P. Mindek[1], J. Sorger[1,2], L. Autin[3], and I. Viola[1]

[1]TU Wien, Austria
[2]VRVis Research Company, Austria
[3] The Scripps Research Institute, La Jolla, California, USA

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 data-centric 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, culling objects are placed in the scene, creating a cutaway visualization of the model. During this process, histograms 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 histograms, which at this point act as interactive visibility equalizers. The technique has been evaluated by domain experts in scientific illustration.

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

1. Introduction

Biology is an emerging field where the state of the current knowledge changes extremely quickly. New discoveries have to be communicated to a large variety of audiences. Since these discoveries often happen on the molecular level and they are not directly observable in sufficient detail, illustration is the only way how to communicate them.

Traditional pipeline of the scientific illustration starts with the collection of data and knowledge gathering. Afterwards, illustrators make sketches, in which specific regions of the illustrated objects are uncovered. For this, occlusion management techniques are necessary. Oftentimes, *cutaway views* are employed, where specific parts of the scene are removed form the organism model in the illustration, so that internal structures become visible. When new knowledge is discovered, the conceptual layout of the illustration might break down and the whole process has to start from the beginning. Therefore, the duration of this process counts in months or even years.

With the rapid changes to the knowledge in the field of biology, it is necessary to adapt the traditional illustration pipeline so that the new data can be easily plugged in and the resulting illustrations can be updated accordingly in a very short time period. Virtual 3D models of cells and other mesoscale molecular structures can be utilized for this purposes. Such models can be created with tools that procedurally assemble individual molecules into large complexes and into entire systems such as bacterial organisms. An example of such a tool is *cellPack* [JAAA*15]. Its core functionality is a packing algorithm, which take individual molecular ingredients, as well as shapes of compartments where the instances of these ingredients are populated, as the input. The output is a 3D model consisting of multiple instances of several molecular ingredients. The instances are densely packed within the predefined compartments. The shape of the compartments is designed by the domain experts in biology.

The mesoscale biological models represent the structure

† Both first authors contributed equally. Contact: {mathieu | mindek}@cg.tuwien.ac.at

of microorganisms, cells, or even viruses at atomic resolution. However, simply displaying such models does not guarantee an adequate view of internal structures. These structures are often the key for the function of the organism, and should be therefore shown in an illustration. The internal structures are occluded because of the high density of the molecular instances present in the models. To solve this problem, visualization techniques need to be developed which reproduce the occlusion management methods used in traditional illustration.

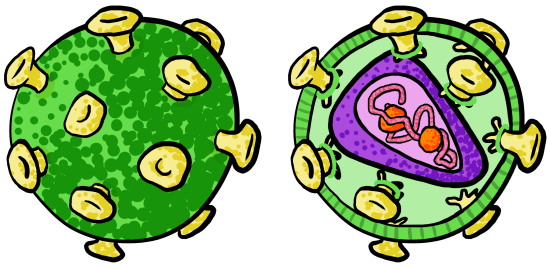
Currently, occlusion management in mesoscale virtual models is carried out by placing clipping objects in the scene, which remove specified parts of the displayed model. During this process, the illustrator does not have a good overview of what instances have been already removed, and which molecular types are still sufficiently represented in the scene. The illustrator has to continuously check the modelled scene against the gathered data and tediously confirm whether all the necessary molecular types are still present.

To alleviate this process, we present our first contribution. During the process of placing the clipping objects in the the scene, we display *visibility histograms* of the molecular types, which immediately reveal which of them are underrepresented or overrepresented. By looking at the visibility histograms, which are continuously updated, the illustrator is able to modify the placement of the clipping objects in such a way that every molecular type is adequately represented in the scene. This is the coarse-level of the visibility specification process.

In traditional illustration, fine-level visibility specification is often utilized as well. To communicate the biology knowledge well, the illustrations have to sometimes display molecular instances which would be impossible to specify with the simple clipping objects, such as cutting planes. An example is shown in Figure 1. Figure 1a shows an illustration of a HIV virus. In Figure 1b, a cutting plane is used to reveal internal structures of the virus - the capsid containing the RNA. Some of the glycoproteins (yellow molecules) are left in the illustration to communicate their presence on the surface of the virus particle. In particular, those glycoproteins which are not occluding the object of interest, were chosen to be kept in the illustration providing the contextual information. 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 pick individual molecular instances to be reintroduced or removed from the scene. This might be done to control the under and overrepresentation of some of the molecular types, removing instances occluding important aspects of the model, suggesting shapes, etc.

To significantly speed up the fine-level visibility specification in our approach utilizing 3D virtual models, we propose our second contribution - *visibility equalizers*. To explain how the visibility equalizers are used to speed up the process



(a) (b)

Figure 1: (a) Illustration of a HIV virus. Here, outside membrane of the virus particle is visible. (b) Cutaway view of the HIV virus. Despite the cutaway, some of the glycoproteins (yellow molecules) are kept in the view to provide adequate context.

of fine-tuning the visibility in molecular models, we use the metaphor of hi-fi sound reproduction. In the hi-fi sound systems, volume control is the basic tool for adjusting the output sound uniformly on all frequencies. This corresponds with the coarse-level visibility specification through clipping objects in the molecular scenes, where all molecular types are uniformly removed from the clipped regions. However, hi-fi sound system allows users to fine-tune the sound through *equalizers*. With equalizers, the volume of each individual frequency band can be adjusted separately to achieve desired sound during the reproduction. To achieve similar level of control for the visibility in the molecular models, we make the visibility histograms interactive. Individual bins of the histograms can be dragged to increase or decrease visibility of the individual molecular types within the scene, given the specified clipping objects. The interactive element effectively turns the visibility histograms into visibility equalizers for the molecular models.

Our main contributions are:

- a new workflow for illustrator-authored cutaway illustrations from mesoscale 3D structural models
- a new visual metaphor of visibility equalizers with which allows users to fine tune the cut-away design so that visibility is distributed among the molecular types as desired by the illustrator.

2. Related Work

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 Finer & Seligmann in 1992 as an automated approach for generating illustrations that consider the occlusion of user defined objects. In 2002, Diepstraten et al [diepstraten2002]picked up the technique again and defined a set of rules for computer-based rendering of technical illustrations to achieve a view-dependent transparency model that mimics the ghosting techniques of technical illustrations. They later extended these rules for interactive cutaway illustrations [diepstraten2003].

Analogous to the cutaways for polygonal representations, WeiÃ§kopf et al. developed an interactive clipping technique for volume rendering that supports complex clipping geometries. In 2004, Viola et al developed an automated approach for focus & context visualization for segmented volumetric objects. An assigned object importance determines the visibility priority for the segmented parts of the volume. contextual information is kept in regions where the context does not occlude the feature of interest. Follow-up work focused on the definition of levels of sparseness and importance compositing for cutaway and ghosting calculations [importance driven feature enhancement]. In 2005, Viola et al. give an overview of current âĂIJsmart visibilityâĂİ techniques that comprise expressive visualization techniques that smartly uncover the most important features of the displayed data, such as cut-away views, ghosted views, and exploded views. baer et al 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. [which part of your phd thesis should I highlight?]

A. KrÃijger et al. combined visualization and interaction techniques such as cutaway views, silhouettes and color-coded distances to improve the spatial perception of feature arrangement for surgical planning. lymph nodes are emphasized using ghosted views to easily convey their spatial position. J. KrÃijger 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 developed an approach that allows interactive exploration of complex models, e.g., mechanical or anatomical, that requires the user to rig each part of the respective model. Based on the rigging, the system produces cuts that adhere to a set of rules that were inspired by anatomic and mechanical illustrations. the approach by burns & finkelstein for view dependent cutaways inspired our aperture that is discussed in section Y. the cutaway shape is determined by the enlarged shape of the focus objects in the depth image. to preserve the information of the cut geometry, they apply shading & contouring/outlining of the cut surfaces, and ghosting of the cut geometry contours. lawonn et al extend this approach to present a composite technique that combines the visualization of blood flow with the surrounding vessel structures. the structures visually encode the wall thickness as colored regions in order to preserve important context information. a view dependent peel-away approach for volume data was proposed by Birkeland and Viola. the approach by Diaz et al preserves the relevant context information in volume clipping by allowing the user to extrude segmented surfaces such as bone structures from the clipping plane.

Sigg et al 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. defined design principles for Cutaway Visualization of Geological Models. they promote boxes as ideal cutaway shapes for emphasizing the shape and depth of focus features in layered structures, such as geological sediments. illumination should effectively communicate the shape and spatial ordering inside the cutaway, as well as enhancing relationships between the focus features and the context. they define five design principles that we discuss in section X in relation to our approach.

Transfer Function Based Approaches. The context-preserving volume rendering model proposed by Bruckner et al is an extension of direct volume rendering. The technique 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 propose a multimodal approach that combines CT scan data and realtime ultrasound data. Importance driven shading is used to emphasize features of higher importance that have been revealed through the culling/ghosting.

The notion of visibility histograms proposed by Correa et al. inspired our visibility equalizer metaphor. These histograms represent the distribution of visibility in a volume-rendered image and should help users manage a set of transfer function parameters to maximize the visibility of interesting intervals in the volume.

Ruiz et al. propose an approach for automatic transfer function optimization. The transfer functions are obtained by minimizing the informational divergence or Kullback-Leibler distance between a user specified target distribution and the visibility distribution captured from certain viewpoints.

[conclusion of the occlusion sub-section] 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

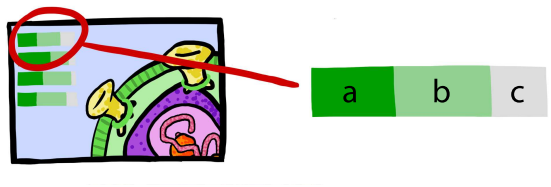


Figure 2: Visual representation of the visibility equalizers.

solid structures that could be made visible with a TF (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 issues 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 fundamentally different from existing occlusion management approaches as it combines principles from object centered and transfer function approaches.

2.2. Visualization of Molecular Structures

The visualization of the molecular structures in our approach is based on the publicly available cellView [le Muzic et al.2015]. The tool is capable of rendering structures that are comprised of several billions atoms at interactive frame rates in multiple levels of detail. Lindow et al. [LBH12] were the ïňĄrst to introduce a fast method for the real-time rendering of large-scale atomic data on consumer grade hardware. Similar to cellView, 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 and eichelbaum et al propose ambient occlusion approaches for large molecular scenes in order to improve the depth perception in these complex structures. Parulek et al 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 large scale molecular visualization, our approach is the first to introduce illustrative occlusion management techniques.

[VKG05] [BHW*07] [BF08] [LRA*07] [LHV12]

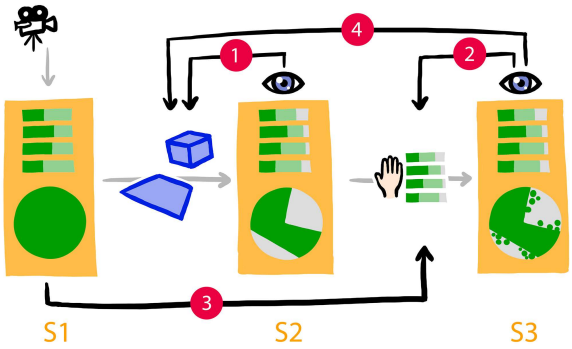


Figure 3: User-centric overview of our method.

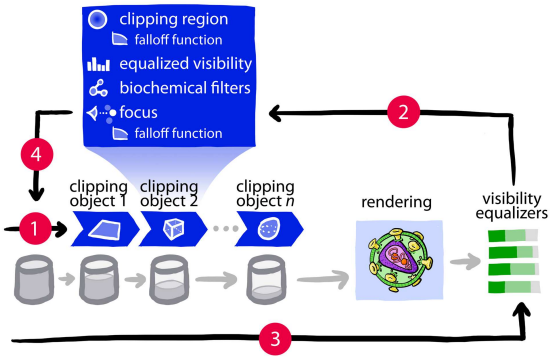


Figure 4: Technical pipeline of our method.

3. Overview

In this work, we focus on visualization and illustration of 3D models of mesoscale molecular structures, such as cells or viruses. We utilize *cellView* [MAPV15] tool for both representation and rendering of our 3D scenes. These scenes can consist up to billions of individual atoms, each belonging to one of the molecular ingredients.

The visual representation of the visibility equalizers is illustrated in Figure 2. The equalizers are represented by three stacked histograms, where each histogram bin represents single molecular ingredients. Each bin is a stack of three values, denoted as *a*, *b*, and *c*. The part *a* shows the number of visible instances of the given ingredient. The part *b* shows the number of instances not removed by clipping, but not visible due to occlusion. The part *c* represents the number of instances which has been clipped away.

Figure 3 illustrates how our method is utilized by users. The method can show the data in one of three states, denoted as *S1*, *S2*, and *S3*. In the state *S1*, the data are shown without any clipping applied. The visibility equalizers are already shown, displaying the proportion of visible and occluded instances of the individual ingredients, however the

user does not interact with them yet. In this state, the user can specify the viewing direction, which might modify the values displayed by the visibility equalizers.

In the state *S2*, parts of the scene are removed. The user can switch from the state *S1* to the state *S2* by placing and manipulating cutting objects. These are represented as distance functions, of which zero level-sets are used as the clipping criteria. The visibility equalizers now also show the portion of the molecular instnaces, which has been clipped away. So far, the clipping has been done in a deterministic manner.

At this point, the user can use the information displayed by the visibility equalizers to steer the iterative process of the placement and the manipulation of the clipping objects (path 1). However, at this point there is also the possibility of the direct interaction with the visibility equalizers. The values in the individual bins of the displayed histograms can be dragged in order to increase or decrease the number of clipped instances of certain ingredients, switching to the state *S3*. This is carried out in a probabilistic manner. If the number of clipped-away instances is decreased, the probability that an instance passing a clipping test will be removed continuously decreases from 100% to 0%. On the other hand, if the the number of clipped-away instances is increased, the probability of instances which do not pass the clipping test being removed from the scene continuously increases from 0% to 100%.

4. Workflow

5. Object-Space Clipping

Clip objects define how many instances of a given ingredient type shall be displayed.

There are two non-exclusive ways how can a clip object influence the clipping, either in object-space or in view-space.

Using an object-space approach, the clip objects will discard instances independently from the view direction.

Clip objects are applied in serial as explained in Figure XX, This operation is recomputed every frame.

In this section we will explain in details how an individual clip object operate for the object-space clipping only.

The first step is the object-space clipping process is the localization of the clip sub-region.

Then, once the clip object is localized, instances will be clipped according to object-space clip parameters which we describe in the following subsections.

Moreover, we introduce advanced parametrization of the distance field falloff, for generating customizable gradient clipping effects.

5.1. Clip-region Localization

Clip objects can be associated with geometrical shapes to localize 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 clip-object geometry.

This operation is repeated every frame, for each instance of the scene.

To determine if an instance lies inside or outside the clip object region, we compute the signed distance between the instance bounding sphere and the closest point on the region surface.

Although supported shapes have a rather simple topology, it may still be computationally expensive, using a mesh-based representation, to compute a signed distance for a large number of instances.

Indeed, using a triangle-based discretization would imply computing the signed distance between the instances and every single triangle of the mesh.

To accelerate the computation we solve the problem analytically using a mathematical description of the 3D signed distance field (SDF).

Using such representation instead reduces the problem of evaluating the signed distance to solving trivial equations.

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 inside of the sphere.

It is worth mentioning that although the set of offered clip-shapes is yet limited it could easily be enriched by utilizing more complex SDF operators, such as union for instance, to merge several shapes together in one single distance field.

5.2. Clip Parameters

A clip-object comprise two basic parameters for each single ingredient that control the visibility of instances, based on their type.

It is worth mentioning that in case no shape is associated with the cull object, the clipping will be evaluated for the entire domain.

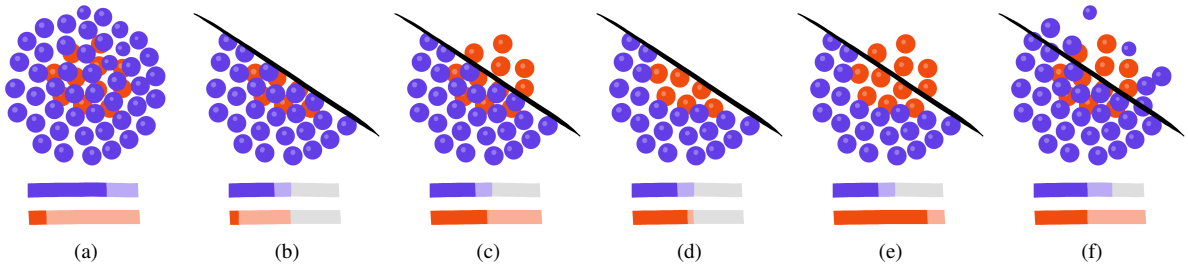


Figure 5: Visibility Equalizers.

The first parameter is the percentage of visible elements of a given type.

We refer to this value as object-space clip probability.

This parameter allows us to control the degree of fuzziness of the clipping.

The other filtering parameters are related to biochemical properties and allows us to control the clipping based on the mass and/or quantity of given ingredient types.

These parameters can be interactively changed via the user interface.

****

Prior to the rendering, after localizing the clip region, each single instance is evaluated in order to determine if it shall be clipped.

First is applied filtering based on the clip probability.

For each instance, we compare a uniformly distributed random number with the clip probability of the instance ingredient type.

If the random number is higher than the probability, the instance is marked as culled, and will not be rendered.

The random number is initially set for each individual instance and remain the same for each re-evaluation of the clipping, 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 range values for the both quantities and molecular weight.

Instances whose properties lie outside on these ranges are marked as culled and will not be rendered.

5.3. Falloff Function

We provide additional parameters to gradually remove instances given a geometrical shape, for illustration purposes.

TODO PMINDEK: Talk about gradient clipping here, motivation and parameters, maybe a figure too.

6. View-Dependent 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 additionally provide a functionality to specify a set of ingredient types as focus, and to selectively remove occluding instances.

We also provide a parameters to control the degree of fuzziness of the clipping.

Finally we introduce a falloff function inspired by our object-space falloff function that allows us to control the degree of aperture of the view-dependent clipping

6.0.1. Occlusion Culling

Due to the potentially large number of instances in our scenes, we accelerate the computation of occluding instances using an image-based approach on the GPU.

Modern graphics hardware already a fixed function called occlusion queries (OQ) and which allow to determine whether an instance is visible or hidden according to previously drawn geometries.

This approach, however, would require issuing one draw call per queries, which can seriously effect the frame rate when issuing hundred of thousands of queries, because of GPU driver overhead.

The rendering tool we are using already allows to render the entire scene in a single call to avoid latency due to the GPU driver.

Therefore, we extend the system using the programmable graphics pipeline instead, to implement custom occlusion queries without GPU driver overhead.

To determine the instances occluding, we priorly render an off-screen texture containing all the focus elements, which we will use as a depth-mask for the occlusion queries.

Instances are rendered using bounding sphere in order to lower to cost of the additional render pass.

Focus ingredients are priorly selected from the user interface.

There can be several ingredient types constituting the focus, however, only one focus mask can be generated per cut object.

Subsequently, we draw the bounding sphere of the remaining instances over the mask, fragments that will pass the depth test are therefore are guaranteed to belong to an occluding object.

From the fragment shader we then update the clip-state of the occluding instance using imageStore() functionality that allows us to write directly to the main video memory.

To determine what instances are in front of the focus, we first separately render a mask containing all the focus elements.

The render pass sets the depth buffer in order to let subsequent draw calls to pass only if they are overlapping the focus region.

Subsequently, we draw the bounding sphere of the remaining instances over the mask, fragments that will pass the depth test are therefore guaranteed to belong to an object occluding the focus, with at least one pixel.

From the fragment program we then mark the occluding instance as culled, in a similar way as we would normally cull an instance.

6.1. Aperture Effect

Jump Flooding Algorithm by Rong & Tan [36]

Image-based mask culling using depth and stencil test

6.2. Island Effect

Image-based mask culling using depth and stencil test

7. Visibility Histograms

To provide a clear overview of the scene properties, we display histograms for each ingredient type that indicate information about their visibility.

By default we chose to show three ranges in each histogram.

The first section of the histogram (dark green region) shows the percentage of instances that are currently visible on the screen.

The entire green section (dark & light green) represents the percentage of instances that are actually rendered.

In order to fill histograms with correct values, we perform book-keeping of both clipped and visible instances, which we recompute every frame.

Histograms are sorted per compartment in a tree layout.

Additional histograms are also displayed for compartments, averaging all the values of the ingredients contained inside.

Upon manipulation of compartment histograms properties, all the children ingredients will be updated accordingly.

Histograms are also interactive.

The user can manipulate histograms individually by dragging the range handles.

Upon manipulation of the right end of the second histogram range (light green) the system will increase or decrease the number of clipped instances.

It is worth mentioning that the user does not directly set the value shown in the histograms, instead he modifies the internal object-space clip probability for the selected cut object and the selected ingredient type,.

, which modified the view and affects histogram values. The internal probability value is hidden to the user to simplify interactions and let the user directly manipulate meaningful values.

The culled states of the instances will get subsequently updated and counted in order to update the histogram value.

Because of the degree of indirection between the user action and the view, we are also able change the way we display information in the histograms, without affecting the way of interacting with them.

For instance, quantities are relative by default, i.e, they represent a percentage, but they can also be displayed as absolute.

For displaying absolute quantities we support logarithmic scaling to ensure low quantities to be visible in the histograms.

An logarithmic ruler is also provided to help the user understanding the displayed values

7.1. Instance Discarding

Prior to the rendering each single instance is evaluated to determine if it shall be rendered.

The cut objects how instances shall be discarded and they are applied sequentially.

Internally the filtering is applied just after the object-space culling as shown in figure XX.

First is applied the filtering based the clip probability.

For each instance, we compare a uniformly distributed random number with the clip probability.

If the random number is higher than the probability, the instance is marked as culled, and will not be rendered.

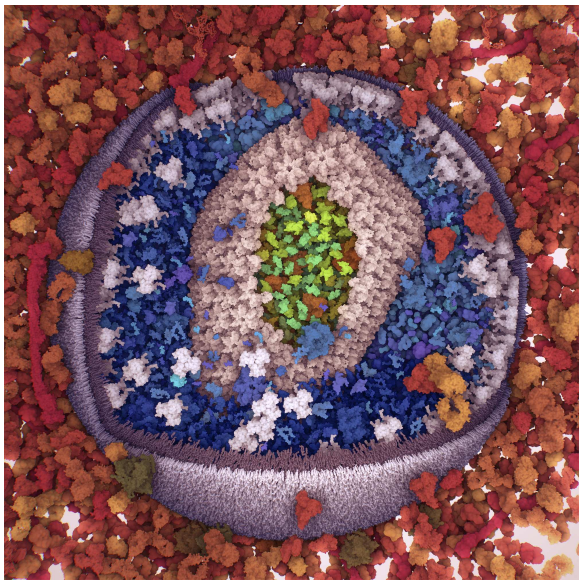


Figure 6: An illustration of the HIV virus in the blood serum utilizing cutaways created with our approach.

The random number is initially set for each individual instance and remain the same, in order to guaranty reproducibility of the scene.

Secondly, instances are filtered according to their biochemical properties, for each cut object and each protein types the user defines ranges values for the both quantities and molecular weight.

Instances whose properties lie outside on these ranges are marked as culled and discarded.

For the book-keeping is the clipped ingredient we count for each ingredient type how many instances where discarded in total, for all active cut object.

8. Depth cues and Enhancements

9. Results and Discussion

10. Evaluation

11. Conclusions

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. 4

[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* (Aire-la-Ville, Switzerland, Switzerland, 2007), EUROVIS'07, Eurographics Association, pp. 275–282. 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. 1

[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. 4

[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. 4

[MAPV15] MUZIC M. L., AUTIN L., PARULEK J., VIOLA I.: cellview: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (Sept. 2015), B"uhler K., Linsen L., John N. W., (Eds.), EG Digital Library, The Eurographics Association, pp. 61–70. 4

[VKG05] VIOLA I., KANITSAR A., GROLLER M. E.: Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics 11*, 4 (July 2005), 408–418. 4