HyperLabels—Browsing of Dense and Hierarchical Molecular 3D Models

David Kouřil, Tobias Isenberg, Barbora Kozlíková, Miriah Meyer, Eduard Gröller, Ivan Viola

HAL Id: hal-02493893

https://hal.inria.fr/hal-02493893

Submitted on 28 Feb 2020

HyperLabels: Browsing of Dense and Hierarchical Molecular 3D Models

David Kouřil, Tobias Isenberg, Barbora Kozlíková, Miriah Meyer, M. Eduard Gröller, Ivan Viola

Abstract—We present a method for the browsing of hierarchical 3D models in which we combine the typical navigation of hierarchical structures in a 2D environment—using clicks on nodes, links, or icons—with a 3D spatial data visualization. Our approach is motivated by large molecular models, for which the traditional single-scale navigational metaphors are not suitable. Multi-scale phenomena, e. g., in astronomy or geography, are complex to navigate due to their large data spaces and multi-level organization. Models from structural biology are in addition also densely crowded in space and scale. Cutaways are needed to show individual model subparts. The camera has to support exploration on the level of a whole virus, as well as on the level of a small molecule. We address these challenges by employing HyperLabels: active labels that—in addition to their annotational role—also support user interaction. Clicks on HyperLabels select the next structure to be explored. Then, we adjust the visualization to showcase the inner composition of the selected subpart and enable further exploration. Finally, we use a breadcrumbs panel for orientation and as a mechanism to traverse upwards in the model hierarchy. We demonstrate our concept of hierarchical 3D model browsing using two exemplary models from meso-scale biology.

Index Terms—Scientific visualization, navigation, 3D molecular data, multi-scale data, hierarchical data, HyperLabels.

✦

1 INTRODUCTION

BIOLOGY as a scientific discipline has reached a point where we begin to understand the essential building blocks of life, on every level of its structural organization. A huge body of work acts as supporting evidence and most of the structural details are now digitally available. This data has sparked an integrative approach to biology: Known elementary pieces of knowledge are combined to describe a more complex system. Visualization plays an important role in this effort [51], [55]. Two such complex systems have been modeled at The Scripps Research Institute using cellPACK procedural modeling [36], i. e., a model of HIV [37] (Figure 1) and a draft model of the Mycoplasma bacterium (Figure 9). While these life forms can be considered rather primitive from a biological perspective, their structural detail results in models with a huge geometric complexity.

This complexity stems from several characteristics of the models, universal for all cellular organisms or viruses. They are hierarchically organized across several *scales*, and each scale contains potentially millions of structural element *instances*. Moreover, these instances are *densely* packed in 3D space. Although we can interactively render huge amounts of geometry today, navigating through and exploring such dense, multi-scale, multi-instance, hierarchical, and three-dimensional scenes remains a fundamental challenge.

The difficulty in navigating these environments is caused by relying on traditional metaphors developed for macroscopic scenery walkthroughs, which often assume scenes composed of extruded 2D surfaces with relatively sparsely populated geometries (e. g., Earth surface). As we deal with entirely new properties in our case, the state-of-the-art navigational metaphors are not well suited for walkthroughs due to the dense, 3D nature of biological scenes.

We address this challenge with a novel navigation technique by leveraging the 3D model's hierarchical meta data. We use the names of substructures of a complex molecular model for annotation as textual labels. These labels—which we call *HyperLabels*—also support user interaction, making them active, clickable elements, as opposed to having just the passive role of annotating visible items that they usually play in visualization. Similar to a click on a folder icon in a file management system, a click on a HyperLabel results in opening up the corresponding structure and revealing its content (i. e., its inner composition). Through this interaction, we allow users to reveal new levels, show previously hidden structures, and make them accessible for further exploration.

This navigation mode of browsing the 3D model through clicking on labels enables navigation both in the spatial space as well as the hierarchically organized scale space. To realize a new form of visual 3D data exploration, we contribute:

- a novel approach for navigating through the hierarchy of 3D structures—the concept of *Hierarchical 3D Model Browsing*, in which a 3D model is recursively explored by selecting individual parts of its hierarchical organization directly in the 3D visualization;
- the concept of *HyperLabels*, i. e., active visualization elements that play both a navigational and annotational role in a visualization system; and
- the design and realization of *Structure of Interest Opening* and its three parts *Sparsification*, *Anchoring*, and *Re-annotation* that deal with scene visibility, camera control, and HyperLabels coordination, respectively.

- D. Kouřil is with TU Wien, Austria. E-mail: dvdkouril@cg.tuwien.ac.at
- T. Isenberg is with Université Paris-Saclay, CNRS, Inria, LRI, France.
- B. Kozlíková is with Masaryk University, Czech Republic.
- M. Meyer is with the University of Utah, USA.
- M. E. Gröller is with TU Wien and the VRVis Research Center, Austria.
- I. Viola is with King Abdullah University of Science and Technology (KAUST), Saudi Arabia.
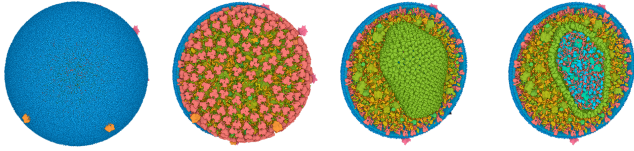
Fig. 1. The organization of models from structural biology can be complex and it is often impossible to convey its architecture using just a single view and cutting settings. Here, we consecutively open up a model of an HIV particle to show its several levels, each communicating a different spatial subpart of the virus.

2 RELATED WORK

In labeling, researchers have previously focused mostly on algorithms that generate optimal label layouts, which we briefly review below. In this discussion we specifically point out examples where labels play more than just a passive role of annotation. Since we use the active, clickable labels for navigation, we also describe how our method relates to previous work in navigation of 3D virtual environments. Finally, the character of biological models presents the challenge of showing all their parts due to the density of such environments, which leads to frequent occlusion. Therefore, we present an overview of occlusion-managing methods.

2.1 (Clickable) Labels in Visualization

Textual labels have been used extensively in scientific visualization to augment the rendered spatial data with a verbal description. Information-rich virtual environments [8] are created, which combine abstract visual representations with 3D virtual environments. For interactive 3D models, emphasis has been on algorithms that generate effective label layouts, as well as temporal coherence upon user interaction. Oeltze-Jafra and Preim [58] provide a comprehensive overview of labeling techniques used in medical data visualization. These techniques can generally be categorized as either external labeling or internal labeling. In external labeling [1], [15], [30], [32], [62], [67], the label text is placed into an empty space, usually around the model, and connected with the structure by a leader line. In internal labeling [5], [14], [60], [64], the label is placed directly over the annotated object. A combination of the two approaches has also been proposed for the specific use in scientific visualization [26], [27], [28]. In addition to these single-scale annotation methods, Kouřil et al. [40] described a method for annotating multi-scale and hierarchical 3D models. Inspired by the level-of-detail techniques in computer graphics, it selects an appropriate level of textual annotation based on the distance to the object.

While in the techniques mentioned so far the labels are mere carriers of information, other approaches facilitate interaction by clicking on a label. Clickable labels are especially prominent in visualizations for anatomy education [63]. As an early example, ZoomIllustrator by Preim et al. [61] provided anatomical models annotated with text on both sides of the 3D view. These annotations were hyperlinks and, by clicking on them, users could reveal more information about particular structures of interest. Later, Mühler and Preim [57] used clickable labels to indicate structures that are invisible as they are occluded by other objects. Upon clicking on a label, the camera moved to a viewpoint from which the

selected structure is fully visible. Jankowski and Decker's dual-mode interface [34] features hyperlink elements that integrate 3D models into a web page environment. In their first mode, the model is embedded into a website, similarly to how it is done with other media (e. g., images). Another mode presents the 3D environment in full screen and integrates the textual annotation into the 3D view. However, they presented their design on a simple scene consisting of just a few geometric primitives such that no changes in the visibility of parts of the environment are needed for exploring it.

Most related to our concept is Lerin et al.'s work [45]. They use labels in a map and clicking on them changes the view position and zoom level based on the clicked structure. They also employ a panel that compiles clicked labels and facilitates hierarchical navigation. They, however, use immediate transitions to a new viewpoint—in contrast to our continuous animation. Moreover, we deal with three-dimensional hierarchical models, which pose many additional challenges, such as objects occluding each other, changing views depending on the camera location, and the need to navigate the space using six degrees of freedom.

The breadcrumbs panel as a hierarchical interaction element can be found also in Kopper et al.'s work [39]. At the top of the screen, they render a miniature of the highest level of the model (i. e., essentially the whole model) and a miniature of the current level of scale. We extend this concept by showing the entire path from the highest level of scale to the current one, thus providing orientation and navigation to all the intermediate scale levels.

2.2 Navigation in 3D Multi-Scale Environments

Several authors discuss aspects of 3D navigation. Jankowski and Hachet [35], for example, surveyed many interaction techniques for navigating 3D environments. In addition, Christie et al. [13] discussed camera control in computer graphics with a focus on semi- and fully-automatic systems. Specifically for visualization systems, they suggest that intelligent algorithms for automatic camera control can take advantage of the data and domain characteristics, as well as the task goals of the user. Both surveys focus on macro-space techniques and ignore multi-scale settings.

Early systems like Pad [59] and Pad++ [4] can be considered as the first multi-scale environments in computing. Furnas and Bederson [25] formalized the concept of 2D multi-scale environments with their Space-Scale Diagrams. Mackinlay et al. [50] provided a solution for efficiently navigating large 3D workspaces by adjusting the camera speed proportionally to the distance from a selected target object. Using this approach, it is possible to skip through the uninteresting (empty) space quickly, while facilitating fine navigation when inspecting a target from close distance. Ware and Fleet [74] extended this concept by changing the camera speed based on a few samples from the current depth buffer. McCrae et al. [52] then developed a technique for navigating multi-scale environments. They use an image-based representation of a scene—a cubemap—that allows them to avoid collisions with scene objects. They demonstrated this approach on a model of the Earth. While such a setting indeed uses multi-scale representations, their scene is not crowded and the objects of interest are mostly distributed on

the surface of a sphere. Tan et al. [68] improved performance of users navigating a 3D environment by speed-coupled flying with orbiting. The speed is coupled with the height and tilt of the camera in respect of the target object. This mode of navigation allowed users to gain a global overview, while still being able to examine local features. The technique is again tailored for a macroscopic world, assuming that all objects reside on a ground surface.

Multiscale navigation is particularly important for visualizations where the data covers multiple orders of magnitude, e. g., in astronomy. OpenSpace [7], for example, relies on the Dynamic Scene Graph [2] to solve the problem of limited floating-point number precision for positions of individual objects. Sagristà et al. [65] described a system for visualizing the star catalog from the Gaia mission, which comprises 1.3 billion stars. To deal with this amount of data, they used a data access mechanism tailored to interactive stellar visualizations as well as an approach ensuring a sufficient precision for high dynamic distance ranges. A recent example of nano-scale visualization is Halladjian et al.'s ScaleTrotter [29] for gene data, where they focus largely on the visual aspects of the scale transitions.

Both astronomical and biological data share the property of scales crossing many orders of magnitude. However, in biology the elements are much more densely packed in space and scale [29]. Individual planets or stars are separated by a vast amount of empty space, whereas the environment of a cell is crowded by omnipresent and always-moving molecules, which we address with our approach.

2.3 Occlusion Management

Due to the dense packing we also need to adjust the scene's visibility settings to prevent occluding objects of interest. Elmqvist and Tsigas [20] define a taxonomy of occlusion management techniques for 3D visualizations, in which they identify five design patterns: multiple viewports, virtual X-ray tools, tour planners, volumetric probes, and projection distorters. They also classify visual tasks affected by occlusion. Among these, the task of *spatial relation*—gaining an understanding of spatial arrangement of several objects—is most relevant in our use case.

Occlusion management approaches can generally be divided into techniques changing optical attributes and techniques modifying the spatial arrangement. Our goal is to inform users about the spatial relation between different parts of a certain biological structure. In this case, it is less desirable to modify the spatial positions of the 3D scene elements. Therefore, we prefer techniques that alter the visual attributes of the structures, e. g., through cutaways or transparency settings, rather than techniques employing exploded views [10], [46], deformations [17], [18], [53], or displacements [21], [22]. Viola and Gröller [72] summarize occlusion-handling approaches in visualization, which address the issue in a *smart* way, i. e., leveraging the underlying features of the model and/or domain as well as the user's expertise. In our work, we leverage the hierarchical and encapsulating character of meso-scale molecular models to automatically create view-dependent cutaways.

For large, multi-scale, multi-instance, dense, and hierarchical *molecular models,* only little work exists that deals with

occlusion management. Most of the visibility techniques work on the scale of individual molecules, like exploded views for showing symmetrical parts of a molecule [70] or depicting possible binding locations for protein interactions [24]. Le Muzic et al. [44] investigated visibility management for complex biological models with an inherent hierarchy to generate static illustrations. Their method is rather tailored for a user-illustrator, who prepares a static illustration that communicates a specific story or character of the model. Sorger et al. [66] used an exploded view on molecular data. In one of their use cases, the technique exposes the inner composition of a structure. However, they also focus on providing an interface (in this case a scripting interface coupled with a visual editor) for a content-creator, rather than a method for interactive exploration.

The computer graphics field offers many cutaway techniques for *polygonal data*, from early investigations of cutaways in interactive settings [23], through the definition of rules for automatic cutaways generation [19], to techniques that give users more control of the shape of the cut [16], [48]. These solutions put emphasis on generating illustrations, yet the demonstrated cases use rather simple models, compared to our molecular models from biology.

Particularly interesting is the work by Li et al. [47], who introduced a sophisticated system for generating cutaways of complex polygonal models. They produce static illustrations that capture the relationships between structures in the proximity of a selected model part. They employ (passive) labels for annotation and allow users to interactively select different parts from a list using a separate UI. Their semi-automatic approach relies on the user categorizing individual parts into types of object shapes to create cuts based on these templates. Instead of such a manual step, in our work we take advantage of the biological model characteristics to generate cutaways fully automatically. Through our HyperLabels concept we also incorporate the multi-scale aspect of the data, while facilitating a more immediate interaction.

Volumetric data (rather than polygonal models), where basically every data point signifies a mass, more closely resemble the density of information of the molecular models we work with. In volume visualization, a transfer function typically maps the sampled value to certain optical attributes, such as color and opacity. A view-dependent approach (rather than a global transfer function for the whole model) is then more suitable for exploratory tasks. A plethora of methods that automatically adjust the visibility setting to gain a view inside the volumetric model exist, many of which attempt to preserve contextual information about the cutaway parts [9], [12], [42].

Viola et al. [73] formalized how less important structures are suppressed to prevent occluding more salient structures in the model. This concept has later also been applied on polygonal data [11]. Building on the importance-driven rendering, Viola et al. [71] presented a framework for an interactive focus of attention for volumetric data, where they automatically compute the most expressive view onto a specific feature. The user selects the focus structure from a list. Employing the underlying importance distribution of features, the visualization gradually changes to shift the attention to the newly selected object. We also select a subset of the model and adjust the visualization to provide
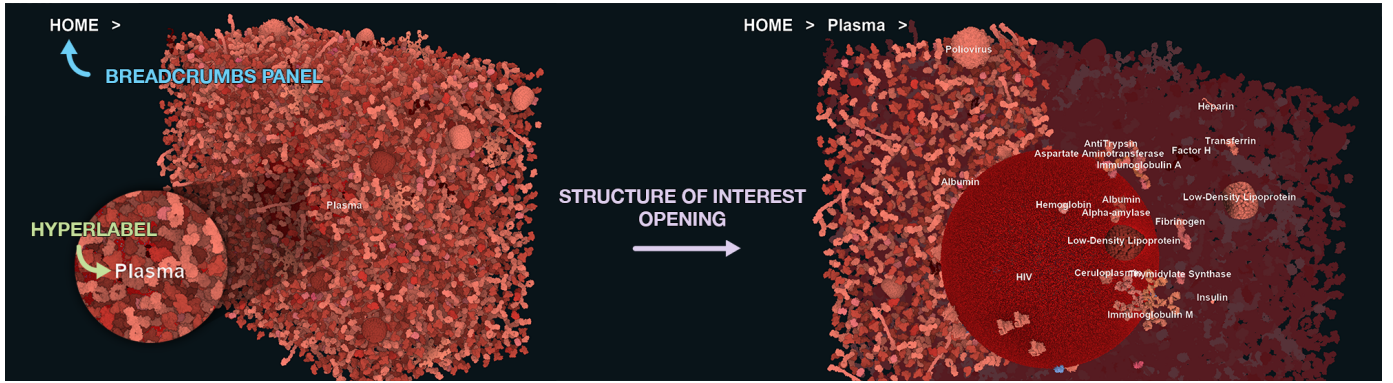
Fig. 2. Overview of the three components of our method: HyperLabels trigger the operation of Structure of Interest Opening, which adjusts the visibility of the model to allow further exploration. This way the model is traversed in a top-down fashion. The breadcrumbs panel provides orientation and enables navigation in the opposite, bottom-up, direction.

a good view onto it. We extend this concept by employing the multi-scale and hierarchical aspects of computationally generated molecular models. We also take advantage of the multi-instance property in resolving occlusions, rather than defining an importance distribution across all scene objects.

3 HIERARCHICAL 3D MODEL BROWSING

As discussed in the last section, existing techniques to interactively explore complex biological models generally suffer from two problems. First, they assume only single-scale environments. Second, they were initially developed for a macroscopic world, i.e., representations of objects approximately the size of a human body. A traditional transition between structures of significantly different sizes, for example, would lead to a frequent use of the zooming functionality due to the multi-scale and dense character of our biologic scenes. It is difficult to set the appropriate zooming speed as it has to be adjusted according to the scale level. For scales representing a whole virus, for instance, the desirable speed is higher than the one for precisely navigating on the scale of small proteins. The multi-scale character of the environment requires changing the speed based on the structure size that is close to the current camera position. While such a method has been proposed, e.g., by McCrae et al. [52], it is highly parameter- and scene-sensitive and can still lead to an extensive zooming to switch between two structures of interest.

We developed our technique for users that are not necessarily experts in biology. As a part of science outreach and education in general, effective and intuitive interactive visualizations are essential for explaining scientific subject matter to a non-expert audience. We leverage the widely used functionality of clicking on textual links that are familiar even to less computer-literate people. In aiming for this target group, expert tasks such as accessing a specific protein from any point in the hierarchy have a lower priority.

We propose a new way of navigating 3D environments that complements existing navigational metaphors employed in visualization like orbiting around the model, panning, and zooming (i.e., flying in the camera direction). We use navigation by direct interaction with the labels. This has the significant benefit that the process of transitioning between two salient objects is automated and requires no further

user interaction. Moreover, we use both the 3D spatial characteristics and the 3D model's hierarchy to facilitate browsing. We borrow the latter from the *click*, *doubleclick*, or *touch* styles of navigation that facilitate intuitive browsing in hypertext documents and file structure trees. Our method for hierarchical 3D model browsing is facilitated by three components in the visualization technique.

First, we deploy **HyperLabels** in the scene to annotate possible targets, i.e., structures of interest (SoIs). Each HyperLabel can be clicked to indicate interest and select the particular object as the target for navigating toward it. Second, in addition to the HyperLabels integrated in the 3D view, we provide a contextual element through a **Breadcrumbs panel**. This panel expresses the current location inside the model's hierarchy and enables selection of a current structure of interest. Third, the selection of a certain label simultaneously also indicates an interest in exploring the inner structure of the object associated to this label. We thus initiate a process of **Structure of Interest Opening** in which the composition of the selected object is revealed in the 3D spatial view.

We realize SoI Opening using three main procedures, which are triggered by clicking on a label. These are: *sparsification*, *anchoring*, and *re-annotation*. We use these procedures in conjunction to facilitate an expressive and seamless browsing of the molecular model. The effect of the three steps is that an object in focus *opens up*, similar to how a folder would open up in a file browser. The user is provided with a glimpse into the inner composition of the selected structure to facilitate understanding of its subcomponents.

Next, we explain the details of our data-driven approach. It takes advantage of the data characteristics but otherwise is fully automatic, without the need for manual pre-processing (such as pre-defining optimal viewing positions). We demonstrate our technique using a scenario from structural biology which not only serves as a motivation for our work but is also a representative setting in which multi-scale and multi-instance structures are featured in a dense, fully three-dimensional environment.

3.1 Abstract and Spatial View Coupling

The meta data containing the information about hierarchical relationships of individual parts of the model represents a

Fig. 3. HyperLabels are textual visualization elements that combine annotation with interaction. Here, HyperLabels are placed in the scene to enable navigation in the spatial model. A blue background is displayed for a HyperLabel hovered over, to indicate, together with the changed cursor, that the text is clickable.
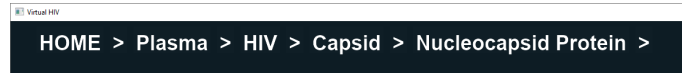

Fig. 4. Inspired by a similar concept used in web design, the breadcrumbs panel provides orientation in the hierarchical organization of a 3D model. Interacting with the breadcrumbs panel facilitates the traversal back, towards higher-level structures in the model. We position the panel in the top left corner of the 3D spatial view, thus integrating a minimal expression of the hierarchical information to the user.

kind of *abstract data* that is usually visualized using abstract visual representations. On the other hand, the 3D model carries information about its spatial characteristics: the spatial arrangement of structures as well as concentrations and distributions of each elementary building type in space. We refer to this second aspect of the model as the *spatial data*. We combine the two by showing the spatial and abstract aspect of the model in a single integrated view, to communicate both facets of the model to the user. We purposefully allocate a bigger part of the screen to the spatial data (as opposed to the other way around; e.g., [3]) to allow viewers to understand how individual parts fit together—a task in which the spatial characteristics matter more than the abstract ones. Our HyperLabels realize a first integration level by placing them directly over the structures in the spatial view, establishing a connection between the shown 3D element and its textual description. With this setup, the user would have to look up the structure in a separate, fully hierarchical view (provided in any arbitrary form) and establish certain relationships between the structure and other objects.

To avoid this separation of both data aspects, we further integrate the spatial and the abstract view using the breadcrumbs panel. Inspired by an established concept from web design [41, Chapter 6], the breadcrumbs panel provides a *path* through the abstract data, showing only its subset that has been traversed. This approach mirrors the same concept used in the spatial view where structures that are not children of the object in focus remain hidden. By employing both of these integration levels, the relevant information about the hierarchical relations of the selected structure is provided to the user directly. We believe that by constructing such a minimal interface we lower the cognitive load, as the user does not have to switch contexts and shift focus between multiple views to get the presently relevant information.

3.2 Top-Down Navigation: HyperLabels

Our HyperLabels naturally fulfill the role of traditional textual labels: they connect the 3D structure with its annotations. We also use them, however, as an entry point for selections. Without our labels, choosing a structure that the user wants to explore next is ambiguous: due to its multi-scale character, a click anywhere in the scene can be interpreted as a selection on any of the levels of the shown structure. Clickable labels thus define a small, specific screen region where a concrete interaction is conducted: Users know exactly which structure they are selecting, and we use this information as an input to the navigation processes that follow.

We see several parallels of HyperLabels with the concept of hyperlinks from the Web. When a hyperlink is clicked, it transports the user to a new location (a web page), and

the content of this new location is shown in the form of text, images, and other multimedia. HyperLabels work in a similar fashion: We use different locations (landmarks) as the destinations to which users can be transported, and the content in our case is the inner structure, i.e., subparts of landmarks. We thus adopted the name as well as some design concepts (e.g., cursor change and different style of rendering upon hover, see Figure 3).

The primary interaction purpose of HyperLabels is, therefore, to establish the upcoming focus: A click on a Hyper-Label selects the corresponding object as the new structure of interest. To fully realize the concept of hierarchical 3D model browsing, however, a label activation has to be coupled with an appropriate follow-up action. To allow users to recursively activate HyperLabel and thus traverse the hierarchy in a top-down direction, we follow the activation by opening of the structure of interest. We thus then show the object's children along with their HyperLabels, which allow users to continue their model exploration further down the hierarchy.

3.3 Bottom-Up Navigation: Breadcrumbs Panel

To also allow users to reverse the direction of their exploration, we could use a simple "back" button, as in Web browsing. This technique would, however, lead to users lose track of their current location in the model hierarchy—an aspect that is of little importance in Web browsing but essential in understanding both the spatial and hierarchical structure of our biological models. We thus employ a breadcrumbs panel (Figure 4), essentially a reduced version of a tree that only shows the path from the current location to the root and thus requires only little screen space [41]. By constructing the breadcrumbs panel from the respective HyperLabels, we allow users to traverse the tree bottom-up, also allowing them to skip several stages if needed.

In our integrated view we link the spatial and abstract parts such that interactions in one part cause changes in the other. We thus ensure that both views remain in sync and show information about the current focus object, along with options for further exploration and the current hierarchy level. The breadcrumbs panel also serves as a kind of URL.

3.4 Structure of Interest Opening (or Closing)

For both ways of navigating the hierarchy, we need a way to adjust the visualization such that it communicates the architecture of the explored subpart and enables further exploration. This opening of the structure of interest comprises of three steps that we describe next.

We cannot continue to show all elements as we explore hierarchy branches due to our models' high density—otherwise only the outer-most layer would be visible. Our

first step is, therefore, to selectively control the visibility of subobject elements using **sparsification** of the dense 3D model. It selects a subset of the model to be shown, based on the focus object. We adjust the visibility settings to provide an unobstructed view into the model that facilitates further exploration of substructures. We describe the specific conceptual sparsification strategies in Section 4.1.

Next, as the new focus object becomes the center of attention, we need to dedicate most of the screen real estate to it. The object may previously have occupied only a few pixels, but to allow users to explore its architectural composition we need to re-allocate the screen space. We thus smoothly adjust the viewpoint and relocate the camera to show the new hierarchy branch's structural representatives that remain after sparsification in the center and in detail. We call this step **anchoring**: It attaches the camera to a particular model subpart and embeds it into its local coordinate system. The anchoring also allows us to use additional, single-scale interaction mechanisms, i. e., orbiting, panning, and zooming. We describe anchoring in Section 4.2.

Sparsification and anchoring change the visualization smoothly, but ultimately a completely new view is shown, with new model elements being visible. In a third step, we thus need to trigger the **re-annotation** of the model to communicate the newly visible landmarks. We describe how to derive the new labels in Section 4.3.

All three procedures—sparsification, anchoring, and re-annotation—result in a change of the visualization. To support viewers in comprehending these changes, we perform each step using an animated transition and in a staged manner [31]. After a click on a scene HyperLabel, we thus first fade out objects due to sparsification, we then transition the camera to its new anchor, and finally fade in new Hy-perLabels. Likewise, for each click in the breadcrumbs panel, we first smoothly zoom out the camera, then fade in the re-appearing objects, and finally show the new HyperLabels.

4 TECHNICAL REALIZATION

The exploration usually starts at the highest level of the hierarchical organization. We thus render the whole 3D model [43] and place HyperLabels corresponding to the highest level of annotation of the spatial view. We initialize the breadcrumbs panel with a home element that allows users to restore the original state of the visualization.

The identifier of an activated HyperLabel now serves as an input for the sparsification. This ID refers to a subpart of the hierarchical organization of the model. There can be many instances of this subpart in the model. In the next step, therefore, we make use of this multi-instance character of the model in adjusting the visibility.

4.1 Sparsification

In contrast to other smart visibility techniques [72] such as cut-aways or exploded views, the main goal of sparsification is to allow users to effectively navigate a dense model—even if the camera is placed in the midst of the model. We thus not only need to remove objects to allow users to focus on the target structure, but we also have to ensure that there is a collision-free camera path toward and, later, for orbiting

around the newly anchored object of interest. Such view manipulations need to be free from suddenly appearing and disappearing structures.

In scene sparsification, we leverage the multi-instance character of the model: Most object types are instantiated in the scene many times (hundred times or more). We can thus maintain the model character and selectively reduce the number of duplicates where they would interfere with the navigation. We carry out the sparsification in two major stages: *Representative Instances Selection* and *Visibility Adjustment*. In the first stage, we select representative instances for all children object types. In the second stage, we resolve the final visibility of these instances. After sparsification, we then pass on the determined visibility settings to the rendering module of the visualization system.

4.1.1 Representative Instances Selection

We select the representative instances, at least one for each object type, both for the level of leaves (e. g., proteins) as well as composite objects (e. g., the HIV particle). We considered different strategies for the selection of the representatives: random selection, first in an array, and closest to the camera. While the computation of the first two is straight-forward, for the latter we compare the distance between the coordinates of the object's center of mass and the camera position in world space. We then select the instance with the shortest such distance as the representative for a particular object type. This approach, in particular, is simple to implement, efficient, and it provides good results in most cases for the used models. More elaborate strategies would also be possible such as mutual distribution in space. Here, distances between the selected representatives could be optimized and we may study variations between optimizing distances in 3D or the projected positions in 2D.

4.1.2 Visibility Adjustment

The most straightforward way of deciding the final visibility is to select exactly those objects that we chose in the previous step as representatives. We call this strategy *all-but-one* spar-sification (Figure 5, top), meaning that all the object instances except one are "sparsified" (i. e., removed). This type of sparsification has the advantage of totally removing the effects of occlusion caused by the scene's density, resulting in a very sparse model. Furthermore, it implicitly ensures that all the representative instances will be observable from some angle.

The disadvantage of all-but-one sparsification is, however, that the real density of the scene is not conveyed in any way after the sparsification and the understanding of the relative abundance of these elementary structures is lost. This is illustrated in Figure 6 where the removed parts are indicated. To counter this weakness, we offer a second strategy—*context-cut* sparsification (Figure 5, bottom)—in which we keep possibly more than one instance visible for each type. We thus show the previously selected representative instances and show them as with all-but-one sparsification. In addition, we place a cutting plane parallel to the viewing plane at the position of the most distant selected representative instance of all types of children and re-introduce all instances that fall behind this cutting plane into the scene. This strategy ensures
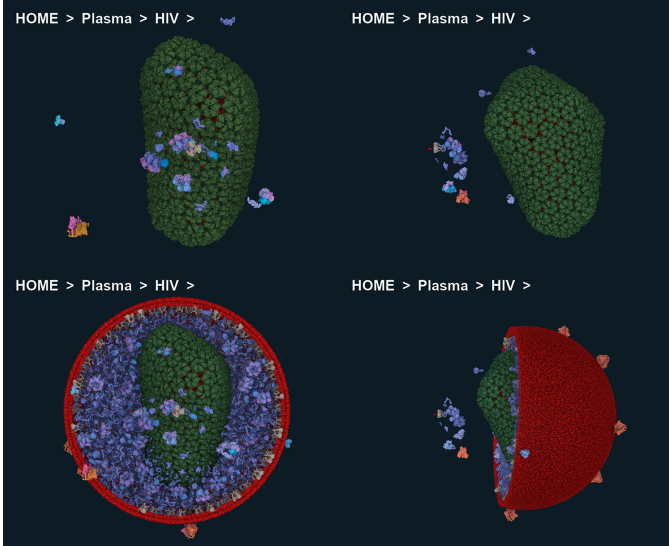
Fig. 5. Comparison of the two implemented sparsification strategies: approach where only exactly one instance for each object on the current level is kept and shown (top); the *context-cut* method where parts of the model behind the farthest representative instance are still shown, by which we provide more contextual information about the spatial organization of the model (bottom).
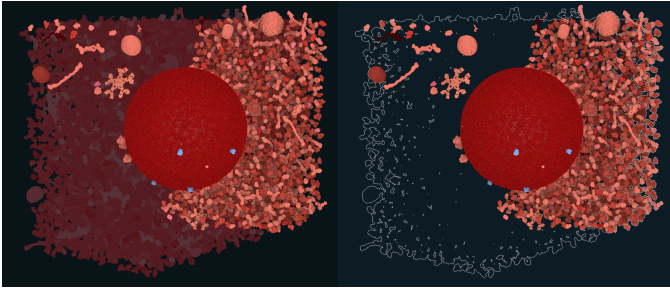

Fig. 6. Parts affected by the sparsification can be rendered with various styles: they can be completely removed from the scene, displayed using a ghosting effect, i. e., rendered with a lower opacity (left), or rendered as only contours (right). Contouring and ghosting indicate the cut-away parts and thus provide a context information to the user.

that all instances for currently shown types are visible, while still keeping the dense context of the model.

We expect that several other sparsification strategies can be found, each working well in specific conditions and scenarios. The strategies that we presented here, however, demonstrate the principle of sparsification as a component in multi-scale model browsing and turned out to be well suited for our examples from biology.

4.1.3 Visual Style

In some cases, the objects indicated as to-be-removed in the sparsification step do not necessarily need to be completely discarded and, for the benefit of the viewer, we might want to keep some as a context. Keeping a certain footprint of the cut-away parts proved to be an important aspect: it communicates the context (e. g., [69]) and users are not misled into thinking that a completely new scene has been shown. We provide more insights on this issue based on an interview with one of our domain experts in Section 6.

To demonstrate this concept (see Figure 6), we implemented a *ghosting* effect [23] which does not completely hide
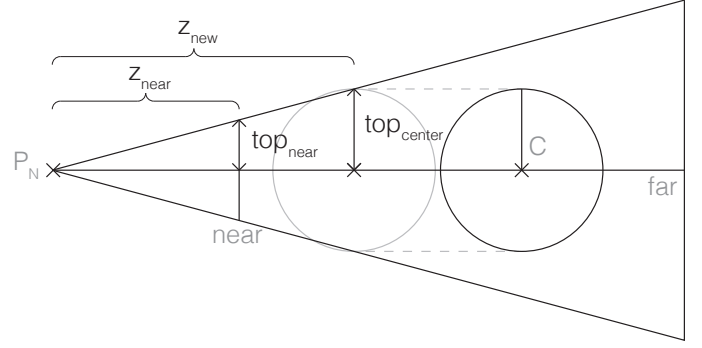

Fig. 7. To show all objects of the current hierarchy level, we enclose them with a bounding sphere. We then move the camera such that the view frustum touches the top (and bottom) points of the bounding sphere.

the cut-away objects, but rather renders them with a lower opacity. We also show a *contour* style (e. g., [6], [33]) where only a line representation of the structure of interest is drawn.

In summary, each scene object belongs to one of three groups depending on the currently selected structure of interest. First, there are those objects that are either representatives or that have been nevertheless kept in the sparsification step. We always render these fully opaque. Second, we have instances of the current object type which have been selected to be cut away. For these objects we can choose the less salient rendering style just discussed. Third, there are those objects which are not relevant in the current view, which belong to a different branch of the hierarchy. We completely remove such objects from the scene.

4.2 Anchoring

The next step brings the newly selected object into focus, moving the camera through the emptied space such that the structure of interest is centered and enlarged. Furthermore, this step re-attaches the camera to this object for future interactions. We thus adjust the camera position, its orientation, and its pivot point. As the possible objects of interest are quite diverse in shape, we start by computing the bounding sphere of the targets on-demand. This process ensures that all selected instances fit into the view, although in the future one may consider to forgo less important structures to prioritize showing more important ones in greater detail.

We then determine the target position of the camera P_N as a point on the line from the bounding sphere's center to the current camera position as follows:

$$P_N = C + z_{new} \cdot |P_C - C|, \tag{1}$$

where P_C is the current camera position, C is the center of the bounding sphere enclosing all shown objects of the new target, and we calculate the value of z_{new} using the camera parameters and the principle of similar triangle ratios (see our illustration of the geometric situation in Figure 7) as:

$$z_{new} = \frac{z_{near} \cdot top_{center}}{top_{near}}. \tag{2}$$

We adjust the orientation of the camera such that its forward vector points in the direction of the center of the current bounding sphere. To ensure that no part of the viewed structure is clipped away by the near plane, we also test whether the closest point on the bounding sphere is

not closer to the camera position than the near plane value. If so, we translate it to be fully inside the volume of the viewing frustum. Last, we set the pivot point for the local orbiting interaction to be at the center of the bounding sphere. This position may seem to be an arbitrary choice because it does not necessarily correspond to any shown object. Consider, however, that the bounding sphere essentially encloses a single object entity at a one-step-higher level of the hierarchy, so essentially we enable users to orbit around the center of this entity and to explore its inner construction.

4.3 Re-annotation

To update the breadcrumbs panel we append the triggered HyperLabel at the end, illustrating the newly shown hierarchical level. To update the set of shown HyperLabels to the new 3D view we use an adjusted version of the Labels on Levels technique [40] for the annotation of multi-scale scenes. In contrast to Labels on Levels, when placing HyperLabels we do not determine the label level (and thus the label shown) automatically using a distance from the camera, but rather provide this information to the algorithm as an input using the output of the sparsification step.

We implement this concept using an *annotation state buffer* that contains the whole scene's labeling state. For all objects it records on which level and with which label this object should be annotated. We update this buffer, for all objects affected by the sparsification, by adjusting their annotation state—labeling them not on the level of the parent (newly selected object in focus) but on the level of the children. We efficiently determine these object children using an object ID look-up table, computed in pre-processing, that contains the immediate children for each object.

The placement of HyperLabels follows the Labels on Levels technique: We select a particular instance from all regions in the image that share the same annotation. We can skip this step for all-but-one sparsification because in that case we only show one object, i. e., the representative instance for each type. Other sparsification strategies (e. g., one that keeps more than one representative instance) provide more candidates for label placement. In that case, we perform sparsification also in the domain of possible labeling candidates, i. e., we render instances but exclude them from the label placement algorithm. We record whether an instance is excluded from the labeling using an additional flag in the *annotation state buffer*. We determine the final position of a HyperLabel as the most salient point on the selected instance projected onto the screen (see the previous publication by Kouřil et al. [40] for further details), and thus overlay a Hyperlabel over the structure. We opted for placing the label directly over the corresponding structure rather than using leader lines, which would introduce additional geometry (and therefore additional visual clutter) into the scene.

We introduce those HyperLabels that are newly added to the scene by fading them in, while those that are removed from the scene are faded out. Later, it is possible for users to explore the focus structure locally using orbiting, panning, or zooming. To facilitate such exploration, we fade out Hyper-Labels during the interaction: Keeping the labels in the scene can distract users as we would constantly have to update their positions. Once the interaction is completed, however, we show the HyperLabels again. After this re-annotation, all HyperLabels are ready for being selected so that the hierarchical exploration of another level can continue.

5 RESULTS

We demonstrate the HyperLabels-enabled spatial and hierarchical browsing using the previously mentioned molecular meso-scale models: a mature *Human Immunodeficiency Virus* (HIV) embedded in blood plasma and a *Mycoplasma genitalium* bacterium. Both models were generated by the cellPACK modeling tool [36] and our proof-of-concept implementation is built on top of the Marion framework [56]. We show the HyperLabels interaction in the supplemental video, where the textual labels can also be observed in a sufficient size.

In the first model, an HIV particle is surrounded by blood plasma proteins, making it initially invisible. The interaction with HyperLabels allows us to reveal the true composition of the blood plasma and show that the virus is submerged in it. We show a tour through the composition of the virus in Figure 8(a)–(c), step-by-step revealing more internal structures of the virus.

The mycoplasma shown in Figure 9 is our second example, it is one of the smallest bacteria. It is only approx. 400 nm long and contains a very small genome, just over 580,000 base pairs, making it a model organism for a minimal genome life form. Besides proteins, it contains many fiber ingredients (RNA, DNA, peptide, and lypoglycane). The depicted model is still a draft, it does not yet contain the bi-lipid membrane that would enclose the internal structure of the bacterium (this will be added in further extensions of the model). Due to its simplicity in compartmentation, the encapsulation relationship is present here as well, although the model is overall less hierarchically structured than the HIV example. Again, the images in Figure 9(a)–(c) show several steps of hierarchical exploration.

We evaluated our navigational technique through a feedback from two experts in molecular and cellular biology. We selected these two experts specifically for their rich background in science communication such as 3D animations as it allows us to understand how our approach is suitable for the education of a general audience. Both of them were familiar with our previous work—one of them is a collaboration partner with whom we have previously worked on a different project. However, neither one of them was involved in the design of the proposed technique.

The first domain expert (22 years of experience in the field) characterized this new way of navigation as a great guided tour with sophisticated clipping, useful for a first-time user getting familiar with a molecular model. It could also be used for presentation purposes (slideshow, screen capture). He noted that our strategy of revealing encapsulated objects automatically generated views that perfectly showcase a particular part of the model (see Figure 8). A previous version we demonstrated to him missed the ability to show the cut-away part using ghosting. He felt that this was a crucial part since, without it, the mental link between what is shown before and after the sparsification could be lost, prompting us to add this feature. Another key observation from him was the fact that a user might not even be aware that labels can be clickable. He brought up the
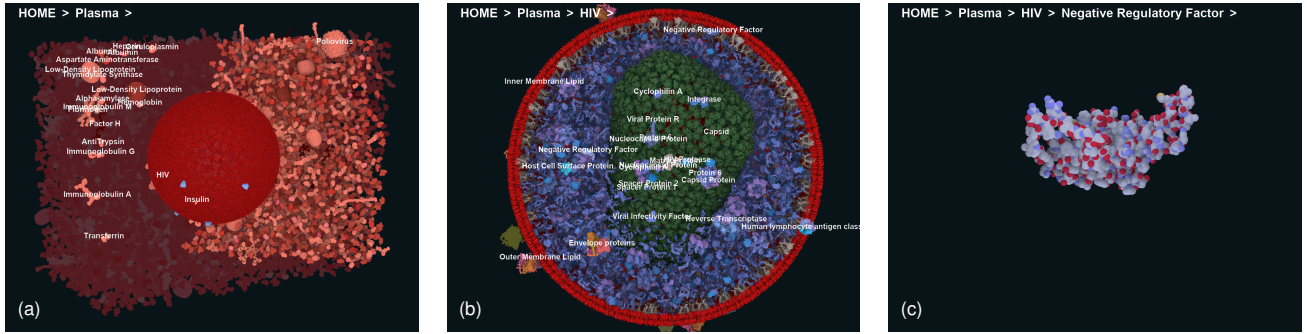
Fig. 8. Example traversal of the HIV in blood plasma model. To see the structure of the virus inside a block of plasma, (a) the Plasma HyperLabel is selected and thus the true composition is revealed, showing the HIV; (b) the hierarchical organization of the HIV model is explored by activating further HyperLabel interaction, (c) ending on the level of a single instance of a viral protein.
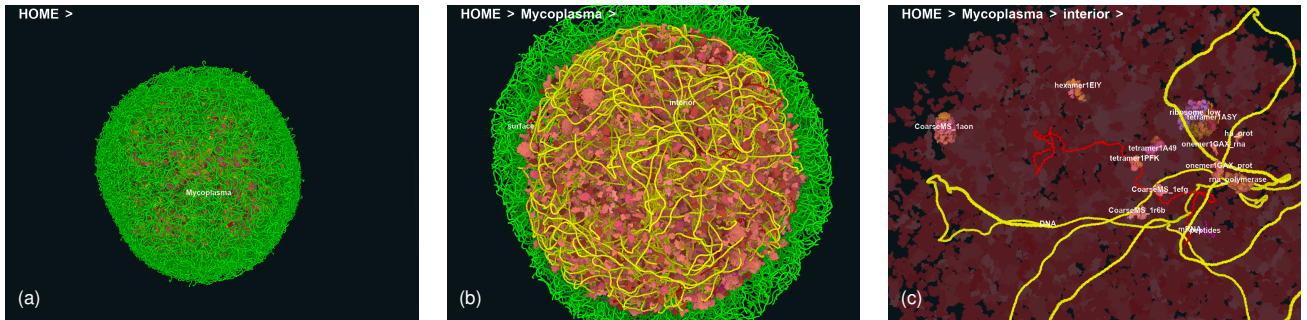

Fig. 9. Example traversal of the Mycoplasma model. The initial view (a) presents the entire model; (b) reveals the inner structure; and (c) shows single instances of each type of molecule in the interior compartment of Mycoplasma.

example of a web link that leads to a change of cursor shape upon hover, which we afterwards also incorporated in the approach. Another possibility he suggested was to make the appearance resemble a button more closely.

The second domain expert (13 years of professional experience), also based on additional feedback from her research group, confirmed the usefulness of our navigation method, especially for students and the general public. She expressed a desire for additional secondary interactions, in this case showing more detailed information about the particular structure and possibly also some way to communicate a sense of the currently depicted size or scale. One member of her group even noted that they would be able to use such a tool as a visual reference because the structures are shown in correct proportions. They also confirmed that sparsification coupled with anchoring can convey a sense of focusing onto a specific part of the model.

6 DISCUSSION

Our new navigational technique is motivated by science dissemination efforts, where scientific concepts—in our case from structural biology—are explained to a broad audience without any significant prior domain knowledge. In this scenario, emphasis has to be put onto the intuitiveness of the exploration experience. A typical end user might be either a member of the general population or a new biology student.

In addition to the possibilities for extension mentioned by the two experts, our approach also has other limitations that we note here. In particular, our method does not allow viewers to jump rapidly between structures from very different

parts of the hierarchical organization. This was, however, a conscious design decision: We wanted to support the task of exploring a hierarchical structure and allow viewers to answer questions about the composition relationship (*"How do the parts of this virus fit together?"*), rather than facilitate the task of random access to a desired object. An additional full hierarchical map in a separate panel or a window could address this issue for environments featuring large or multiple screens. This additional needed screen space could, however, present a concern in other settings (e. g., immersive science center installation). Alternatively, we could also expand the breadcrumbs panel only on demand (e. g., after pressing a button) into a full tree view, through which the user can select any structure of the model.

Finally, even though we allow users to explore models with a complexity that, up to recently, was impossible to interactively manage, we still only cover a relatively small sub-set of the scale space. Models that, for example, start with macroscopic organisms (e. g., a human being) and contain detail down to the cell chemistry are still impossible to render today and may require us to adjust our hierarchical exploration concept. Moreover, not all multi-scale building blocks are defined with a *containment* relationship—the DNA, for example, requires other multi-scale exploration strategies [29] we would need to incorporate into our concept.

We set out to solve a problem of navigation under the—from the point of navigation—previously not considered constraints of multi-scale, multi-instance, hierarchical, and dense 3D models. Re-implementing the method on other models with these characteristics should be straightforward.

In addition, we believe that it will be useful for any dense and hierarchical 3D model even if it does not fulfill the rest of the conditions. In that case, however, adjustments need to be conducted, tailored for the specific need, e. g., using a different smart visibility technique [72] instead of sparsification for models that do not have multiple instances.

7 CONCLUSION AND FUTURE WORK

In the past, interactive 3D model exploration for visualization and other domains has been possible only either in terms of spatial navigation (most 3D visualization tools support this aspect) or in terms of understanding the hierarchy of multi-scale models (e. g., [46], [54]). Both aspects are essential and many innovative solutions have been presented for them to date. In our work, to the best of our knowledge for the first time, we allow users to navigate both facets of the data in a single, integrated interaction concept. In contrast to existing smart visibility techniques [72] and single-stage hierarchy exploration [61], we support, and put the emphasis on, the interaction for traversing the hierarchical organization directly. Our approach handles the spatial model as faithfully as possible, not treating the encapsulating structures just as layers to be clipped away in order to show the internals. The definition of what is the object in focus (and what is the context that is to be visually suppressed) is fluid in our method, depending on which structural level is currently relevant to the user. In fact, with HyperLabels the choice to jump between these levels is absolutely in the hands of the user. With this approach we allow users to understand—in particular for applications in biology—how parts are assembled into structural complexes. With our new interactive approach we thus allow viewers to learn both about the 3D structure, the names of the elements, and the structural composition of complex subject matter. We thus see the main application of our approach in education, yet applications are possible wherever multi-scale containment and spatial structure needs to be portrayed simultaneously.

Regarding future research, a logical next step would be to perform a user study with inquiry into the usability of the presented technique. Answering the questions of usability, intuitiveness, and effectiveness in various practical scenarios, however, might be more relevant for an HCI venue.

In addition, the three low-level procedures of our proposed technique can be, in principle, replaced by other implementations, perhaps specific for a certain domain-specific use case. For example, there might be better, more semantically-driven ways to realize anchoring, e. g., along the lines of the HoverCam technique [38]. Furthermore, interactive sparsification, where the visibility settings of the scene is adjusted even upon camera movement in the anchored state, should be investigated. Here, the main challenge is maintaining temporal coherence. Solutions such as Lidal et al. [49] can serve as an inspiration.

Finally, we are considering to explore the concept of *procedural* HyperLabels where the annotation is not concerned with a structure, but with annotation of a process on which the structure is participating. Once the label is selected, the particular process is initiated, while the process as well as the structure can be viewed at various levels of detail.

REFERENCES

[1] K. Ali, K. Hartmann, and T. Strothotte, "Label layout for interactive 3D illustrations," *Journal of the WSCG*, vol. 13, no. 1, pp. 1–8, 2005. doi: 11025/1441

[2] E. Axelsson, J. Costa, C. Silva, C. Emmart, A. Bock, and A. Ynnerman, "Dynamic scene graph: Enabling scaling, positioning, and navigation in the universe," *Computer Graphics Forum*, vol. 36, no. 3, pp. 459–468, Jun. 2017. doi: 10.1111/cgf.13202

[3] J.-P. Balabanian, I. Viola, and E. Gröller, "Interactive illustrative visualization of hierarchical volume data," in *Proc. Graphics Interface*. Toronto: CIPS, 2010, pp. 137–144.

[4] B. B. Bederson and J. D. Hollan, "Pad++: A zooming graphical interface for exploring alternate interface physics," in *Proc. UIST*. New York: ACM, 1994, pp. 17–26. doi: 10.1145/192426.192435

[5] B. Bell, S. Feiner, and T. Höllerer, "View management for virtual and augmented reality," in *Proc. UIST*. New York: ACM, 2001, pp. 101–110. doi: 10.1145/502348.502363

[6] P. Bénard and A. Hertzmann, "Line Drawings from 3D Models: A Tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 11, no. 1–2, pp. 1–159, Sep. 2019. doi: 10.1561/0600000075

[7] A. Bock, E. Axelsson, K. Bladin, C. Jonathas, P. Gene, T. Matthew, J. Kilby, K. Masha, C. Emmart, and A. Ynnerman, "OpenSpace: An open-source astrovisualization framework," *Journal of Open Source Software*, vol. 2, no. 15, p. 281, Jul. 2017. doi: 10.21105/joss.00281

[8] D. A. Bowman, C. North, J. Chen, N. F. Polys, P. S. Pyla, and U. Yilmaz, "Information-rich virtual environments: Theory, tools, and research agenda," in *Proc. VRST*. New York: ACM, 2003, pp. 81–90. doi: 10.1145/1008653.1008669

[9] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller, "Illustrative context-preserving exploration of volume data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1559–1569, Nov./Dec. 2006. doi: 10.1109/TVCG.2006.96

[10] S. Bruckner and M. E. Gröller, "Exploded views for volume data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1077–1084, Sep./Oct. 2006. doi: 10.1109/TVCG.2006.140

[11] M. Burns and A. Finkelstein, "Adaptive cutaways for comprehensible rendering of polygonal scenes," *ACM Transactions on Graphics*, vol. 27, no. 5, pp. 154:1–154:7, Dec. 2008. doi: 10.1145/1457515.1409107

[12] M. Burns, M. Haidacher, W. Wein, I. Viola, and E. Gröller, "Feature emphasis and contextual cutaways for multimodal medical visualization," in *Proc. EuroVis*. Goslar, Germany: Eurographics Assoc., 2007, pp. 275–282. doi: 10.2312/VisSym/EuroVis07/275-282

[13] M. Christie and P. Olivier, "Camera control in computer graphics: Models, techniques and applications," in *SIGGRAPH ASIA Courses*. New York: ACM, 2009, pp. 3:1–3:197. doi: 10.1145/1665817.1665820

[14] G. Cipriano and M. Gleicher, "Text scaffolds for effective surface labeling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1675–1682, Nov./Dec. 2008. doi: 10.1109/TVCG.2008.168

[15] L. Čmolík and J. Bittner, "Layout-aware optimization for interactive labeling of 3D models," *Computers & Graphics*, vol. 34, no. 4, pp. 378–387, Aug. 2010. doi: 10.1016/j.cag.2010.05.002

[16] C. Coffin and T. Höllerer, "Interactive perspective cut-away views for general 3D scenes," in *Proc. 3DUI*. Los Alamitos: IEEE Computer Society, 2006, pp. 25–28. doi: 10.1109/VR.2006.88

[17] C. Correa, D. Silver, and M. Chen, "Illustrative deformation for data exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1320–1327, Nov./Dec. 2007. doi: 10.1109/TVCG.2007.70565

[18] C. D. Correa, D. Silver, and M. Chen, "Discontinuous displacement mapping for volume graphics," in *Proc. Volume Graphics*. Goslar, Germany: Eurographics Assoc., 2006, pp. 9–16. doi: 10.2312/VG/VG06/009-016

[19] J. Diepstraten, D. Weiskopf, and T. Ertl, "Interactive cutaway illustrations," *Computer Graphics Forum*, vol. 22, no. 3, pp. 523–532, Sep. 2003. doi: 10.1111/1467-8659.t01-3-00700

[20] N. Elmqvist and P. Tsigas, "A taxonomy of 3D occlusion management for visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, pp. 1095–1109, Sep./Oct. 2008. doi: 10.1109/TVCG.2008.59

[21] N. Elmqvist, "Balloonprobe: Reducing occlusion in 3D using interactive space distortion," in *Proc. VRST*. New York: ACM, 2005, pp. 134–137. doi: 10.1145/1101616.1101643

[22] N. Elmqvist and M. Eduard Tudoreanu, "Occlusion management in immersive and desktop 3D virtual environments: Theory and evaluation," *International Journal of Virtual Reality*, vol. 6, no. 2, pp. 21–32, Jun. 2007.

[23] S. K. Feiner and D. D. Seligmann, "Cutaways and ghosting: Satisfying visibility constraints in dynamic 3D illustrations," *The Visual Computer*, vol. 8, no. 5–6, pp. 292–302, Sep. 1992. doi: 10.1007/BF01897116

[24] K. Furmanová, J. Byška, E. M. Gröller, I. Viola, J. J. Paleček, and B. Kozlíková, "COZOID: Contact zone identifier for visual analysis of protein-protein interactions," *BMC Bioinformatics*, vol. 19, no. 1, pp. 125:1–125:17, Apr. 2018. doi: 10.1186/s12859-018-2113-6

[25] G. W. Furnas and B. B. Bederson, "Space-scale diagrams: Understanding multiscale interfaces," in *Proc. CHI*. New York: ACM, 1995, pp. 234–241. doi: 10.1145/223904.223934

[26] T. Götzelmann, K. Ali, K. Hartmann, and T. Strothotte, "Adaptive labeling for illustrations," in *Proc. Pacific Graphics*, 2005, pp. 64–66.

[27] T. Götzelmann, K. Ali, K. Hartmann, and T. Strothotte, "Form follows function: Aesthetic interactive labels," in *Proc. CAe*. Goslar, Germany: Eurographics Assoc., 2005, pp. 193–200. doi: 10.2312/COMPAESTH/COMPAESTH05/193-200

[28] T. Götzelmann, K. Hartmann, and T. Strothotte, "Agent-based annotation of interactive 3D visualizations," in *Proc. Smart Graphics*. Berlin, Heidelberg: Springer, 2006, pp. 24–35. doi: 10.1007/11795018_3

[29] S. Halladjian, H. Miao, D. Kouřil, M. E. Gröller, I. Viola, and T. Isenberg, "Scaletrotter: Illustrative visual travels across negative scales," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, Jan. 2020, to appear. doi: 10.1109/TVCG.2019.2934334

[30] K. Hartmann, K. Ali, and T. Strothotte, "Floating labels: Applying dynamic potential fields for label layout," in *Proc. Smart Graphics*. Berlin, Heidelberg: Springer, 2004, pp. 101–113. doi: 10.1007/978-3-540-24678-7_10

[31] J. Heer and G. Robertson, "Animated transitions in statistical data graphics," *IEEE Transaction on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, Nov./Dec. 2007. doi: 10.1109/TVCG.2007.70539

[32] Z.-D. Huang, S.-H. Poon, and C.-C. Lin, "Boundary labeling with flexible label positions," in *Proc. Algorithms and Computation*. Cham, Switzerland: Springer International Publishing, 2014, pp. 44–55. doi: 10.1007/978-3-319-04657-0_7

[33] T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte, "A developer's guide to silhouette algorithms for polygonal models," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 28–37, Jul./Aug. 2003. doi: 10.1109/MCG.2003.1210862

[34] J. Jankowski and S. Decker, "A dual-mode user interface for accessing 3D content on the world wide web," in *Proc. WWW*. New York: ACM, 2012, pp. 1047–1056. doi: 10.1145/2187836.2187977

[35] J. Jankowski and M. Hachet, "A survey of interaction techniques for interactive 3D environments," in *Eurographics State of the Art Reports*. Goslar, Germany: Eurographics Assoc., 2013, pp. 65–93. doi: 10.2312/conf/EG2013/stars/065-093

[36] G. T. Johnson, L. Autin, M. Al-Alusi, D. S. Goodsell, M. F. Sanner, and A. J. Olson, "cellPACK: A virtual mesoscope to model and visualize structural systems biology," *Nature Methods*, vol. 12, no. 1, pp. 85–91, Dec. 2015. doi: 10.1038/nmeth.3204

[37] G. T. Johnson, D. S. Goodsell, L. Autin, S. Forli, M. F. Sanner, and A. J. Olson, "3D molecular models of whole HIV-1 virions generated with cellPACK," *Faraday Discussions*, vol. 169, pp. 23–44, Sep. 2014. doi: 10.1039/c4fd00017j

[38] A. Khan, B. Komalo, J. Stam, G. Fitzmaurice, and G. Kurtenbach, "Hovercam: Interactive 3D navigation for proximal object inspection," in *Proc. I3D*. New York: ACM, 2005, pp. 73–80. doi: 10.1145/1053427.1053439

[39] R. Kopper, T. Ni, D. A. Bowman, and M. Pinho, "Design and evaluation of navigation techniques for multiscale virtual environments," in *Proc. VR*. Los Alamitos: IEEE Computer Society, 2006, pp. 175–182. doi: 10.1109/VR.2006.47

[40] D. Kouřil, L. Čmolík, B. Kozlíková, H. Wu, G. Johnson, D. S. Goodsell, A. Olson, M. E. Gröller, and I. Viola, "Labels on levels: Labeling of multi-scale multi-instance and crowded 3D biological environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 977–986, Jan. 2019. doi: 10.1109/TVCG.2018.2864491

[41] S. Krug, *Don't Make Me Think: A Common Sense Approach to the Web*, 2nd ed. Thousand Oaks, CA, USA: New Riders Publishing, 2005.

[42] J. Krüger, J. Schneider, and R. Westermann, "ClearView: An interactive context preserving hotspot visualization technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 941–948, Sep./Oct. 2006. doi: 10.1109/TVCG.2006.124

[43] M. Le Muzic, L. Autin, J. Parulek, and I. Viola, "cellVIEW: A tool for illustrative and multi-scale rendering of large biomolecular datasets," in *Proc. VCBM*. Goslar, Germany: Eurographics Assoc., 2015, pp. 61–70. doi: 10.2312/vcbm.20151209

[44] M. Le Muzic, P. Mindek, J. Sorger, L. Autin, D. Goodsell, and I. Viola, "Visibility equalizer: Cutaway visualization of mesoscopic biological models," *Computer Graphics Forum*, vol. 35, no. 3, pp. 161–170, Jun. 2016. doi: 10.1111/cgf.12892

[45] P. M. Lerin, D. Yamamoto, and N. Takahashi, "Making a pictorial and verbal travel trace from a GPS trace," in *Proc. W2GIS*. Berlin, Heidelberg: Springer, 2012, pp. 98–115. doi: 10.1007/978-3-642-29247-7_9

[46] W. Li, M. Agrawala, B. Curless, and D. Salesin, "Automated generation of interactive 3D exploded view diagrams," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 101:1–101:7, Aug. 2008. doi: 10.1145/1360612.1360700

[47] W. Li, L. Ritter, M. Agrawala, B. Curless, and D. Salesin, "Interactive cutaway illustrations of complex 3D models," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 31:1–31:11, Aug. 2007. doi: 10.1145/1276377.1276416

[48] R. H. Liang, G. J. Clapworthy, M. Krokos, and R. Mayoral, "Real-time predefined shape cutaway with parametric boundaries," in *Proc. CGIV*. Los Alamitos: IEEE Computer Society, 2005, pp. 227–231. doi: 10.1109/CGIV.2005.64

[49] E. M. Lidal, H. Hauser, and I. Viola, "Design principles for cutaway visualization of geological models," in *Proc. SCCG*. New York: ACM, 2012, pp. 47–54. doi: 10.1145/2448531.2448537

[50] J. D. Mackinlay, S. K. Card, and G. G. Robertson, "Rapid controlled movement through a virtual 3D workspace," *ACM SIGGRAPH Computer Graphics*, vol. 24, no. 4, pp. 171–176, Aug. 1990. doi: 10.1145/97880.97898

[51] X. Martinez, M. Krone, N. Alharbi, A. S. Rose, R. S. Laramee, S. O'Donoghue, M. Baaden, and M. Chavent, "Molecular graphics: Bridging structural biologists and computer scientists," *Structure*, vol. 27, no. 11, pp. 1617–1623, Nov. 2019. doi: 10.1016/j.str.2019.09.001

[52] J. McCrae, I. Mordatch, M. Glueck, and A. Khan, "Multiscale 3D navigation," in *Proc. I3D*. New York: ACM, 2009, pp. 7–14. doi: 10.1145/1507149.1507151

[53] M. J. McGuffin, L. Tancau, and R. Balakrishnan, "Using deformations for browsing volumetric data," in *Proc. Visualization*. Los Alamitos: IEEE Computer Society, 2003, pp. 401–408. doi: 10.1109/VISUAL.2003.1250400

[54] H. Miao, E. De Llano, J. Sorger, Y. Ahmadi, T. Kekic, T. Isenberg, M. E. Gröller, I. Barišić, and I. Viola, "Multiscale visualization and scale-adaptive modification of DNA nanostructures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 1014–1024, Jan. 2018. doi: 10.1109/TVCG.2017.2743981

[55] H. Miao, T. Klein, D. Kouřil, P. Mindek, K. Schatz, M. E. Gröller, B. Kozlíková, T. Isenberg, and I. Viola, "Multiscale molecular visualization," *Journal of Molecular Biology*, vol. 431, no. 6, pp. 1049–1070, Mar. 2019. doi: 10.1016/j.jmb.2018.09.004

[56] P. Mindek, D. Kouřil, J. Sorger, D. Toloudis, B. Lyons, G. Johnson, M. E. Gröller, and I. Viola, "Visualization multi-pipeline for communicating biology," *IEEE Transactions on Visualization and*

Computer Graphics, vol. 24, no. 1, pp. 883–892, Jan. 2018. doi: 10.1109/TVCG.2017.2744518

[57] K. Mühler and B. Preim, "Automatic textual annotation for surgical planning," in *Proc. VMV*.   DNB, 2009, pp. 277–284.

[58] S. Oeltze-Jafra and B. Preim, "Survey of labeling techniques in medical visualizations," in *Proc. VCBM*.   Goslar, Germany: Eurographics Assoc., 2014, pp. 199–208. doi: 10.2312/vcbm.20141192

[59] K. Perlin and D. Fox, "Pad: An alternative approach to the computer interface," in *Proc. SIGGRAPH*.   New York: ACM, 1993, pp. 57–64. doi: 10.1145/166117.166125

[60] R. D. Prado and A. B. Raposo, "Real-time label visualization in massive CAD models," in *Proc. CAD/Graphics*.   Los Alamitos: IEEE Computer Society, 2013, pp. 337–344. doi: 10.1109/CADGraphics.2013.51

[61] B. Preim, A. Raab, and T. Strothotte, "Coherent zooming of illustrations with 3D-graphics and text," in *Proc. Graphics Interface*. Toronto: CIPS, 1997, pp. 105–113. doi: 10.20380/GI1997.12

[62] B. Preim, A. Ritter, T. Strothotte, T. Pohle, D. R. Forsey, and L. Bartram, "Consistency of rendered images and their textual labels," in *Proc. CompuGraphics*.   Queluz, Portugal: GRASP, 1995, pp. 201–210.

[63] B. Preim and P. Saalfeld, "A survey of virtual human anatomy education systems," *Computers & Graphics*, vol. 71, pp. 132–153, Apr. 2018. doi: 10.1016/j.cag.2018.01.005

[64] T. Ropinski, J.-S. Prassni, J. Roters, and K. Hinrichs, "Internal labels as shape cues for medical illustration," in *Proc. VMV*.   DNB, 2007, pp. 203–212.

[65] A. Sagristà, S. Jordan, T. Müller, and F. Sadlo, "Gaia Sky: Navigating the Gaia catalog," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1070–1079, Jan. 2019. doi: 10.1109/TVCG.2018.2864508

[66] J. Sorger, P. Mindek, P. Rautek, M. E. Gröller, G. Johnson, and I. Viola, "Metamorphers: Storytelling templates for illustrative animated transitions in molecular visualization," in *Proc. SCCG*. New York: ACM, 2017, pp. 27–36. doi: 10.1145/3154353.3154364

[67] T. Stein and X. Décoret, "Dynamic label placement for improved interactive exploration," in *Proc. NPAR*.   New York: ACM, 2008, pp. 15–21. doi: 10.1145/1377980.1377986

[68] D. S. Tan, G. G. Robertson, M. Czerwinski, and M. Czerwinski, "Exploring 3D navigation: Combining speed-coupled flying with orbiting," in *Proc. CHI*.   New York: ACM, 2001, pp. 418–425. doi: 10.1145/365024.365307

[69] C. Tietjen, T. Isenberg, and B. Preim, "Combining silhouettes, shading, and volume rendering for surgery education and planning," in *Proc. EuroVis*.   Goslar, Germany: Eurographics Assoc., 2005, pp. 303–310, 335. doi: 10.2312/VisSym/EuroVis05/303-310

[70] M. Trellet, N. Férey, M. Baaden, and P. Bourdot, "Content-guided navigation in multimeric molecular complexes," in *Proc. BIOSTEC*.   Portugal: SCITEPRESS, 2014, pp. 76–81. doi: 10.5220/0004914300760081

[71] I. Viola, M. Feixas, M. Sbert, and M. E. Gröller, "Importance-driven focus of attention," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 933–940, Sep./Oct. 2006. doi: 10.1109/TVCG.2006.152

[72] I. Viola and M. E. Gröller, "Smart visibility in visualization," in *Proc. CAe*.   Goslar, Germany: Eurographics Assoc., 2005, pp. 209–216. doi: 10.2312/COMPAESTH/COMPAESTH05/209-216

[73] I. Viola, A. Kanitsar, and M. E. Gröller, "Importance-driven volume rendering," in *Proc. Visualization*.   Los Alamitos: IEEE Computer Society, 2004, pp. 139–146. doi: 10.1109/VISUAL.2004.48

[74] C. Ware and D. Fleet, "Context sensitive flying interface," in *Proc. I3D*.   New York: ACM, 1997, pp. 127–130. doi: 10.1145/253284.253319

David Kouřil is a doctoral student at TU Wien, Austria. He received his Master's degree (Mgr.) in 2017 from Masaryk University in Brno, Czech Republic. His research topic lies in scientific visualization, where he focuses on three-dimensional data coming from structural biology and designs novel visualization and interaction methods that support exploration and understanding of the environments that this data represents.

Tobias Isenberg is a senior research scientist at Inria, France. He received his doctoral degree from the University of Magdeburg, Germany, in 2004. Previously he held positions as post-doctoral fellow at the University of Calgary, Canada, and as assistant professor at the University of Groningen, the Netherlands. His research interests comprise topics in scientific visualization, illustrative and non-photorealistic rendering, and interactive visualization techniques. He is particularly interested in the benefit, use, and control of abstraction for illustrative visualization.

Barbora Kozlíková is an Associate Professor at the Faculty of Informatics, Masaryk University in Brno, Czech Republic. Her research interests include visualization and visual analysis of biomolecules, with focus on comparative and explorative tasks in large molecular dynamics simulations. She is the head of the VisIt Lab at the Faculty of Informatics: the visualization group specializing on diverse topics in visualization and its connections to machine learning and virtual reality.

Miriah Meyer is an associate professor in the School of Computing at the University of Utah and a faculty member in the Scientific Computing and Imaging Institute. She co-directs the Visualization Design Lab, which focuses on the design of visualization systems for helping people make sense of complex data, and on the development of methods for helping visualization designers make sense of the world. She obtained her bachelors degree in astronomy and astrophysics at Penn State University, and earned a PhD in computer science from the University of Utah. Prior to joining the faculty at Utah Miriah was a postdoctoral research fellow at Harvard University and a visiting scientist at the Broad Institute of MIT and Harvard.

M. Eduard Gröller is professor at TU Wien, Austria, and adjunct professor of computer science at the University of Bergen, Norway. His research interests include computer graphics, visualization, and visual computing. He became a fellow of the Eurographics Association in 2009. He is the recipient of the Eurographics 2015 Outstanding Technical Contributions Award and of the IEEE VGTC 2019 Technical Achievement Award.

Ivan Viola is an Associate Professor at King Abdullah University of Science and Technology (KAUST), Saudi Arabia. In his research, Viola investigates methods that automate the visualization design process. Basal new technologies he then applies to various application domains, such as ultrasound diagnostics, geology, or submilimeter biology. Viola has graduated from TU Wien, Austria in 2005 and moved for postdoctoral fellowship to the University of Bergen, Norway, where he was gradually promoted to the rank of Professor. In 2013 he has been awarded a Vienna Science and Technology Fund grant to establish his own research group back at TU Wien, Austria. After several years of research on visualization and computer graphics techniques for multiscale biological structures in Vienna, Viola has co-founded the startup company called Nanographics, to commercialize nanovisualization technologies. Currently, he has relocated to KAUST for taking advantage of generous research conditions to multiply his investigative efforts, for delivering technologies that make visual, in-silico life possible.