TU WIEN Informatics

Interactive Visualization of Dense and Multi-Scale Data for Science Outreach

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Mgr. David Kouřil
Matrikelnummer 01529693

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assoc. Prof. Dipl.-Ing. Dr.techn. Ivan Viola
Zweitbetreuung: Ao. Univ. Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Diese Dissertation haben begutachtet:

Bernhard Preim	Kwan-Liu Ma

Wien, 19. Februar 2021

David Kouřil

TU WIEN Informatics

Interactive Visualization of Dense and Multi-Scale Data for Science Outreach

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Mgr. David Kouřil
Registration Number 01529693

to the Faculty of Informatics

at the TU Wien

Advisor: Assoc. Prof. Dipl.-Ing. Dr.techn. Ivan Viola
Second advisor: Ao. Univ. Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

The dissertation has been reviewed by:

Bernhard Preim	Kwan-Liu Ma

Vienna, 19th February, 2021

David Kouřil

Erklärung zur Verfassung der Arbeit

Mgr. David Kouřil

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 19. Februar 2021

David Kouřil

Acknowledgements

It is my great honor to be submitting this thesis after five years with the VisGroup. I have experienced many ups and downs, which made it into a poweful and lasting memory. Here I would like to mention people who shared this life part with me and contributed to my doctoral studies.

For starters, I'd like to thank the two "gateway" people, without who I would not have gone to do a PhD. First, I would like to thank Barbora Kozlíková. Without her, I would not have gone to Vienna for an Erasmus stay, and I would not have eventually gone to do a PhD with the VisGroup. I'm happy that I approached her when choosing what to do and that she turned from a teacher into a friend over the years. Next, I thank Mathieu Le Muzic, who I first met when he was visiting at Masaryk University. Working with him for the few months, I've probably learned more about computer graphics than in my whole computer graphics curriculum.

Next, I wish to thank my PhD supervisor, Ivan Viola. He took me under his wings, and I could count on his support, knowing he was doing the maximum for me to succeed. Without his strong supervision and visionary thinking, none of this work would be possible. Meister Eduard Gröller, as my de-facto co-supervisor, contributed with many crazy ideas and even crazier paper titles. I hope I fulfilled at least some of your visions as a thank you for the research environment that you formed in the group.

I believe that attempting hard things can generate the greatest camaraderie and build lifelong friendships. For me, the PhD was at many points a struggle. Even if I gained no degree from these five years, it still would have been worth it just because of the friendship I developed with Haichao Miao. Now Dr. Haichao Miao has helped me tremendously not only in academic work but also in life in general. He truly made Vienna a second home for me.

I want to thank all the members of the VisGroup that I shared the timeline with: Renata, Hsiang-Yun, Tobi, Peter, Aleksandr, Manu, and Johannes. I have a special affection for Renata since she came to the group at roughly the same time as me, and her kind character was a great pleasure to be around. Special mention goes to Jirka Chmelík, with whom I shared a room when Vienna was a scary new place to me. I would like to wish the best of luck to the next generation of PhD students who started close to the end of my run: Laura and Nicholas.

I am happy to have the chance to collaborate with excellent international scientists. It is unbelievable that I got to work with people like Art Olson, David Goodsell, Ludo Autin (Scripps), or Graham Johnson (Allen Institute). I would like to specifically thank three external people who helped me when I most needed it. For my first paper, I owe a lot to Ladislav Čmolík, who provided incredible support in the last-minute writing up of the manuscript. In the second paper, Tobias Isenberg proved himself to be an absolutely brilliant collaborator, and I learned a lot from him. Finally, in the third paper, Ondra Strnad was a huge help not only with the workload but also with his calm and "above-it" approach. Furthermore, it was great fun working with Sarkis Halladjian and advice him on his PhD, even though he finished it before me.

The most personal thanks come last. For many years I've had the greatest supporter in my girlfriend, Gabriela. Thank you, Gabi. I dedicate this thesis to you for standing by my side through all of it.

In many ways, I do not consider any successes in my life to be just mine, but primarily that of my parents. They've worked incredibly hard to build a good life for my sister and me. I could not have gone from a life in a two-thousand people village to being on an international level of a field without their support. My thanks, therefore, belong to my mum Lenka and my dad Václav.

Kurzfassung

Die Fortschritte im Rendering wissenschaftlicher Daten haben es Forschern ermöglicht, immer umfangreichere und komplexere 3D-Modelle zu visualisieren. Insbesondere bei der Visualisierung biologischer Daten auf der Nanoskala hat sich die Herausforderung langsam vom Echtzeit-Rendering auf die Probleme im Zusammenhang mit dem Verständnis und der Erkundung komplexer virtueller Umgebungen verlagert, die aus Modellen bestehen, die potenziell aus mehreren Millionen Atomen bestehen können. In Anbetracht der Tatsache, dass wissenschaftliche Visualisierung nun zunehmend dazu dient, den aktuellen Stand der Wissenschaft an die breite Öffentlichkeit zu kommunizieren, wird es immer wichtiger, Technologien zu entwickeln, die es den Menschen ermöglichen, mit den Visualisierungen zu interagieren.

Diese Arbeit konzentriert sich auf das Problem der Navigation in komplexen 3D-Modellen, die aus einer großen Anzahl von molekularen Instanzen bestehen, die dicht im dreidimensionalen Raum gepackt sind. Aufgrund der großen Umgebung, die über mehrere Größenordnungen erstreckt, sind die traditionellen Navigationsparadigmen, die in der Echtzeit-Computergrafik eingesetzt werden, bei der Anwendung auf biologische Umgebungen unzureichend.

Im ersten Teil der Arbeit analysiere ich die Herausforderung der Navigation, die genau ein solcher Anwendungsfall mit sich bringt, und beschreibe mehrere Navigationsmodi, die bei der Interaktion eines komplexen 3D-Visualisierungssystems mit dem Endbenutzer eingesetzt werden können. Diese Navigationsmodi werden implementiert, indem navigationsbezogene Parameter der Visualisierung für den Endbenutzer dargestellt werden. Alternativ wird die Kontrolle über diese Parameter vom Visualisierungssystems algorithmisch berechnet. Wir diskutieren drei solcher Modi der Navigation: augmentiv, deklarativ und automativ. Erstens: Bei der augmentiven Navigation erhält der Benutzer die volle manuelle Kontrolle über alle Aspekte der Navigation, wie z. B. die Steuerung der Kameraposition und -rotation oder die Sichtbarkeit der einzelnen Teile des Modells. Seine manuelle Erkundung wird jedoch durch automatisch bereitgestellte Beschriftungen ergänzt, die dem Benutzer helfen, sich in der Umgebung zurechtzufinden. Bei der deklarativen Navigation wird die Interaktion des Benutzers auf die Angabe seines Ziels vereinfacht. Das Visualisierungssystem übernimmt dann die Low-Level-Steuerung der Visualisierung, z. B. die Animation des Kamerapfads und den Übergang der Sichtbarkeit der Szene, die so berechnet werden, dass der Benutzer direkt zu seinem angegebenem

Ziel navigiert wird. Die dritte Stufe, die automatische Navigation, entlastet den Benutzer sogar von dieser Verantwortung und überlässt die Wahl dessen, was gezeigt wird, einer algorithmischen Lösung. In diesem Fall kann ein solcher automatischer Fly-Through dann durch eine bestimmte Storyline geführt werden.

Im zweiten Teil der Arbeit stelle ich konkrete Methoden vor, die technische Lücken adressieren und die im ersten Teil vorgestellten Navigationsschritte zu realisieren. Ich beginne mit der Einführung einer Methode zur textuellen Beschriftung von multiskalare Molekülmodellen, inspiriert durch das Level-of-Detail-Konzept. Auf diese Weise wird ein Szenario der augmentativen Navigation bereitgestellt. Zweitens schlage ich eine Navigationsmethode zur Durchsuchen eines dichten molekularen Modells mit hierarchischer Organisation vor, die das deklarative Navigationskonzept umsetzt. Die vorgestellte Methode verwendet textuelle Beschriftungen zum Durchsuchen des dreidimensionalen Modells und bietet im Wesentlichen eine Möglichkeit, die hierarchische Organisation zu durchlaufen und die räumlichen Eigenschaften des Modells zu erkunden. Abschließend schlage ich eine Pipeline zur Erstellung automatisierter Touren durch molekulare Modelle vor und demonstriere damit den automatisierten Navigationsmodus.

Abstract

The progress in rendering scientific data has enabled researchers to visualize progressively more extensive and complex 3D models. Specifically, in biological visualization of the nano-scale, the challenge has slowly shifted from real-time rendering onto the issues related to understanding and exploring complex virtual environments built up by models consisting of, potentially, tens of millions of atoms. Considering that scientific visualization now increasingly serves to communicate the current knowledge to the general public, it is becoming more important to develop technology that allows people to interact with the visualizations.

This thesis focuses on the problem of navigating complex 3D models composed of large numbers of molecular instances packed densely in the three-dimensional space. Due to the large environment encompassing several magnitudes of scale, the traditional navigational paradigms applied in real-time computer graphics are becoming insufficient when applied to biological environments.

In the first part of the thesis, I analyze navigation challenges presented by such a use case and recognize several modes of navigation that can be employed when interfacing a complex 3D visualization system with the end-user. These navigational modes are implemented by exposing navigation-related parameters of the visualization to the end-user. Alternatively, control over these parameters is entrusted to the visualization system side and determined algorithmically. We discuss three such modes of navigation: augmentive, declarative, and automative. First, in augmentive navigation, the user is given fully manual control over all aspects of navigation, such as controlling the camera's position and rotation, or the visibility of the individual model's parts. Their manual exploration is, however, augmented by automatically deployed annotation to help make sense of the environment. In declarative navigation, the user interaction is simplified to declaring their target. The visualization system then takes over the low-level controls of the visualization, e.g., camera path animation and scene visibility transition, which are computed to navigate the user directly to their declared target. The third stage, automative navigation, relieves the user from even this responsibility and places the choice of what gets shown to an algorithmic solution. In this case, such automated fly-through can then be guided by a specific storyline.

In the second part of the thesis, I present specific methods addressing technical gaps and contributing to realizing the navigational stages presented in the first part. I start by

introducing an approach for textual labeling of multi-scale molecular models inspired by the level-of-detail concept. That way, a scenario of augmentive navigation is provided. Second, I propose a navigational method for traversing a dense molecular model with a hierarchical organization, implementing the declarative navigation concept. The presented method uses textual labels for browsing the three-dimensional model, essentially providing a way of traversing the hierarchical organization and exploring the spatial characteristics of the model. Finally, I propose a pipeline for producing automated tours of molecular models, demonstrating the automative navigation mode.

Contents

CHAPTER 1

Introduction

Biology has, over the years, served as a rich source of data for the visualization community. These two fields have developed a strong interdisciplinary collaboration, exemplified by the conferences such as EG VCBM, BioVis, or VIZBI. The biological field has benefited on several occasions from the involvement of visualization experts who work on turning raw data into visual representations and leveraging the powerful processing of the human visual system [MKA+19].

Visualization methods are particularly beneficial for studying biological phenomena on the smallest scales, which are often the subject of molecular and cellular biology subfields. A human eye cannot observe an environment on such scales, and instruments are required to gain insight into life processes on this level. Visualization then comes up as a natural candidate for turning the data outputted by imaging and resolving instruments into a visual representation that can be more easily processed by people. Scientific visualization functions as an extension of human senses, enabling us to perceive otherwise unseen structures and processes. Molecular and cellular biology have previously been divided by the sheer scale difference and insufficient instrumentation to observe both using the same methods at one time. Lately, however, through modern computing methods, these two biology branches are being connected. Scientists can now acquire biological models that capture much more of the living organisms, both in space and in scale. The result is models that capture several magnitudes of scale at the same time. In this thesis, we work with several of these *multi-scale models*.

In the thesis, we contribute technical solutions for biology communication. We base our techniques on the interactive visualization of data provided by biology researchers and develop methods for presenting this data's interesting features to a broad audience. As a result, we do not serve a single specific user or a user group. Our work instead contributes to technology that can be beneficial to a whole spectrum of users. In our thinking, we do, however, emphasize an audience with very little domain knowledge. We imagine the venues where science communication happens these days: museums, science centers, open

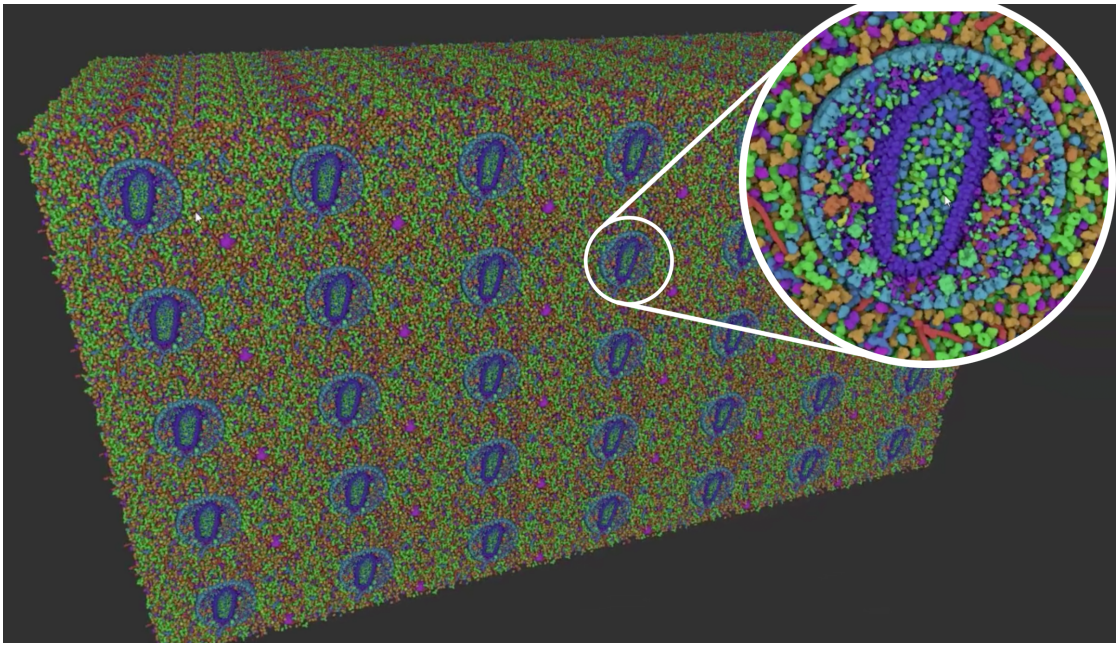


Figure 1.1: Example of an environment built up by large molecular models. This artificial scene has been created by copying model of an HIV embedded in blood plasma several times in order to test rendering capabilities of the cellVIEW system [LMAPV15]. It contains roughly $15 \cdot 10^9$ atoms.

doors days at research institutions, or—these days more than often—the Internet. We aim to capture the attention of people who might not even have a previous interest in the domain matter. Another example is a beginning student who possesses some domain knowledge but is still in need of guidance. To help us evaluate our contributions aimed at such a variety of users, we often collaborated with domain experts. Our collaborators are not only experts in biology, but they are scientists also very well-versed in science communication. Therefore, we leverage their knowledge and rich experience of both of these aspects in evaluating our techniques.

While we cannot yet actually assemble a virtual model of a cell with all its components down to atomistic detail, we can already see the issues with large molecular models from experimentation with smaller biological structures, e. g., viruses. The term large in this context refers to the size and scale range at the limits of what can be rendered with standard consumer graphics hardware at interactive framerates. In Figure 1.1, we see a visualization of an artificial dataset created to test out the rendering capabilities of the cellVIEW system [LMAPV15]. It is built up by copying an actual dataset of an HIV particle embedded in blood plasma several times. The resulting scene contains $15 \cdot 10^9$ atoms. We will not describe how to render such large molecular models in this thesis. Le Muzic's work [Le 16] offers a deep dive into rendering complex molecular scenes efficiently using modern GPU methods. We will instead turn the attention to the

problem of presenting these visualizations to the end-user. More specifically, we will look at methods for making it easier for a general audience member to comprehend, navigate, and explore visualizations like the example above.

We can observe several things from Figure 1.1. First, on this magnification level, the individual atoms are, by far, not the recognizable building element of the scene. Instead, one may identify individual molecular instances, or even the viral particles, to be the elementary type on this level. Second, to truly communicate the build-up of such a large and complex scene, we need to address the omnipresent occlusion caused by the model's crowdedness. The outer layer must be clipped away to showcase a biological structure's inner components, i. e., what lies hidden inside the "cube". The occlusion causes problems also when the virtual camera is immersed in the model, which leads to the user essentially "not seeing the forest for all the trees". However, from an even more fundamental perspective, we start by addressing the fact that our target users might not have any idea of what they are watching. We aim the visualizations in this thesis to be consumable by people with a wide range of experience. Only a small percentage of people will recognize specific biological structures, e. g., the human immunodeficiency viruses in Figure 1.1, from just their shapes.

The thesis is primarily based on three research works, which will be presented later. While they each solve a specific scientific visualization problem, they all contribute to enabling users to *navigate* complex virtual environments resulting from modeling biological structures. By visualizing progressively larger biological structures, we create virtual worlds so extensive that we need additional tools to traverse this space efficiently. The need for sophisticated navigation tools is even more prominent considering our intention of science communication. When interfacing such a complex visualization with a user with little to no previous domain knowledge, they must have ways of not getting lost in the unknown environment.

Typically in a visualization pipeline, dozens, if not hundreds, of parameters in each stage influence the final visual output. It is infeasible to expose all those parameters to the end-user. Some user groups can tolerate a broader parameter range (e. g., experienced domain experts or people with experience with visualization systems), but other groups can get confused by the sheer amount of options. In the thesis, we mostly focus on parameters related to navigation through the virtual environment, which we term *navigational degrees of freedom*. The degree of freedom term is used in several scientific fields, e. g., mechanics or statistics, referring to a parameter in a system that is allowed to change. Such change leads to a different configuration of the system. In mechanics, one can imagine a robotic arm where each joint influences the final position of the arm's tip. Each of the joints contributes to one or more degrees of freedom of the whole system's movement.

We will apply the term *navigational degrees of freedom (NDoFs)* for the navigation parameters in a 3D visualization. Each adjustable component of the visualization pipeline represents one degree of freedom of the navigation that the user (or the system) can control in the exploration. As an example, one elementary component used in almost every 3D visualization is a virtual camera. We will consider the camera one of the

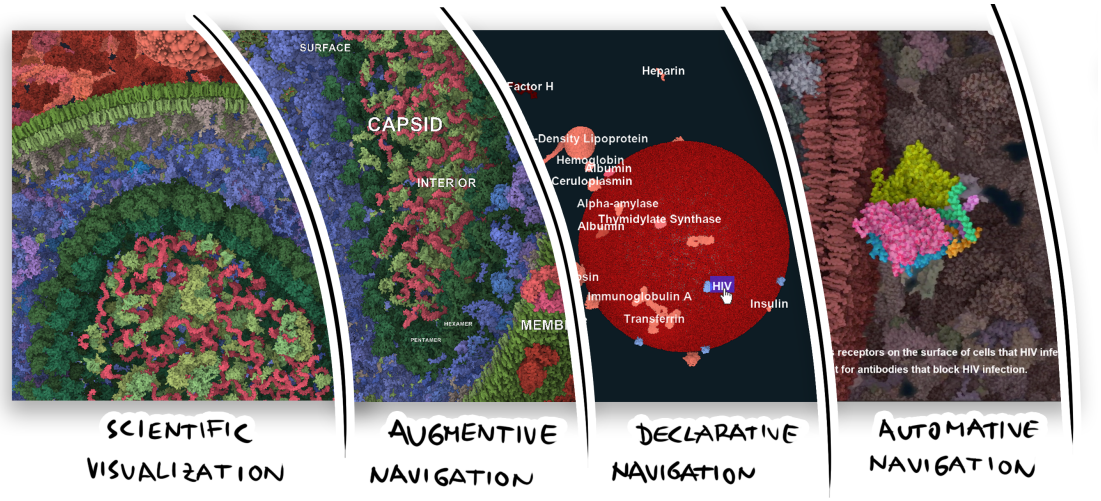


Figure 1.2: Illustrating the navigational modes: The three modes—augmentive, declarative, and automative—provide three levels of navigation used to explore a the 3D environment of a scientific visualization.

NDoFs. Naturally, even a simple virtual camera has potentially 6 degrees of freedom in the traditional sense: three positional parameters (x-, y-, and z-coordinates) and three rotational parameters (rotation along the camera's x-, y-, z-axis). Other adjustable camera parameters may include near and far planes, a field of view, or focal distance. Previous research on simplifying camera movement has focused mostly on stabilizing each of these low-level DoFs. For example, automatizing the camera path generation and reducing the movement to the control of a single degree of freedom by, for example, scrolling with the mouse wheel [MMGK09].

In this thesis, we propose a paradigm that instead stabilizes the high-level degrees of freedom, i. e., reducing the number of degrees of freedom like the complete camera control to simplify navigation in a 3D visualization. We thus define *navigational modes* (Figure 1.2) where different high-level DoF are either placed in the control of the human user, or as an alternative, given as a responsibility to an algorithmic approach on the visualization system's side. In Chapter 3, we specifically talk about three such navigational modes: *augmentive*, *declarative*, *automative*.

In the *augmentive* mode, the user has complete manual control over the exploration of the 3D model. While this allows for full control over the visual output, it also means that the user bears the responsibility of finding the right parameter settings to gain insight from the data's visual representation. However, we assist the user with such tasks by augmenting the spatial visualization with annotation, orientation, and guidance elements. This mode can be compelling for users who have a certain insight into the models' domain and can recognize salient structures from the less important ones. The *declarative* mode is more tailored for persons that do not have a deep understanding of the domain matter and run the potential risk of getting completely lost when let to explore by themselves.

In this mode, the low-level controls of the camera path and adjusting the visibility of the model components are entrusted to a procedural approach inside the visualization system. The user explores the environment by specifying what they wish to see, i. e., they declare their intended target, and the visualization automatically takes them to this target. In the third, *automative*, mode, the user is relieved even from the responsibility of choosing their targets. In venues such as museums, science centers, or conference exhibitions, direct one-on-one user interaction can be undesirable. With the automative mode, the visualization can provide engaging visuals even without a user choosing its targets. The sequence of presented objects can be chosen algorithmically or even follow a specific storyline.

Exploration has become an overloaded word in the world of visualization, to the point where some researchers try to refrain from using it as a task description [Ada17], aiming instead for more specific terms to describe the end goals for which their visualization tools are designed. In scientific visualization, however, where the spatial characteristics are still one of the essential features of the visualized data, we believe this keyword is still relevant. Walking, or rather flying, through a virtual environment built up by actual measured data can help scientists put concepts together, think holistically, and potentially devise new hypotheses. With our research work described in this thesis, we contribute towards bridging explanatory and exploratory techniques into eventually fully functioning models of structures that build up life. We essentially work towards creating a powerful microscope enabled by interactive computer graphics, allowing people to access, observe, and explore environments on scales that are otherwise unseeable.

1.1 Research Goals and Contributions

The research conducted and presented in this thesis set the following research goals:

1. *Efficiently and effectively include textual labels in complex virtual environments made up of biological models.* Incorporating textual labels in the scientific visualization of spatial data enables users to make sense of the depicted environments. Characteristics of these spatial models, however, prohibit implementing previously proposed labeling techniques. Novel methods must be designed to include labels in multi-scale multi-instance hierarchical and dense 3D molecular models.

2. *Enable intuitive navigation through hierarchically-organized dense 3D models.* Information organized in hierarchies has been prominent in information visualization, and many methods for navigating such an organization have been proposed and implemented. However, navigating three-dimensional structures that are hierarchically organized has previously not been investigated to such an extent. Visualizing 3D models inherently leads to the occlusion of some of their parts. The occlusion must be handled to properly communicate a model's components and allow navigating its structural organization.

3. *Automatically generate virtual tours of complex molecular structures.* While it helps when an audience member can move through the hierarchical composition of the 3D model, without an expert description, the many exciting stories in the data are not communicated. In some scenarios, requiring direct user interaction can also be undesirable. Therefore, we aim to generate virtual tours of the model, which take the audience on a guided journey through the model. Such an approach presents an opportunity for incorporating storylines authored by domain experts.

The core contributions of this thesis present technical solutions to achieve the research goals mentioned above. We describe these techniques in Chapters 4, 5, and 6.

Chapter 4 presents a conceptual framework for labeling in 3D molecular environments with the multi-scale, multi-instance, dense, and hierarchical characteristics. We continue by proposing a real-time algorithm that chooses the appropriate label level for each object in the current view and selects appropriate representative instances of each object type for the label placement. The placement algorithm is coupled with animated transitions triggered by changes in the label layout to ensure that the labeling is temporally coherent during user interaction.

Chapter 5 proposes a novel approach for navigating through the hierarchy of 3D structures by so-called *browsing*. A 3D model is recursively explored by selecting individual parts of its hierarchical organization directly in the 3D visualization. Textual labels are promoted into an active role and become clickable *HyperLabels*. They are used to select the current structure of interest. The selection leads to an *opening* operation that adjusts scene visibility, camera control, and annotation to reveal the selected structure in the dense and hierarchical 3D model.

Chapter 6 first outlines the conceptual *Scalable Documentary* framework, which comprises real-time methods for producing scientific documentaries in a scalable and future-proof way. *Molecumentaries*, i. e., molecular documentaries using multi-scale, multi-instance, and dense 3D molecular models, are then used as an exemplary application of the scalable documentary concept. We present an automated method for gathering descriptional information about the model components and constructing the story structure from these descriptions into a story graph. We then describe a method for synthesizing a specific narrative by traversing the story graph, procedurally generating automatic cinematic animations, and supplying the visuals with corresponding verbal commentary generated using text-to-speech technology.

As a natural progression from the technically-oriented research conducted throughout my doctoral studies, the final contribution of this thesis comprises a three-level conceptual framework for navigation in scientific visualization. The *navigational modes* framework is described in Chapter 3. The framework is defined as a combination of *navigational degrees of freedom* placed either in the user's hands or entrusted to an algorithmic solution of the visualization system itself.

1.2 Thesis Overview

This thesis comprises research work done in the author's doctoral program between the years 2017 and 2020. It consists of three separate research projects, each resulting in a scientific manuscript. Two of those manuscripts have been published at the highest venue of the visualization research field. The first publication, Paper A, received a Best Paper Honorable Mention Award at the VIS 2018 conference. The two publications serve as a base material for Chapter 4 and Chapter 5, respectively:

Paper A: D. Kouřil, L. Čmolík, B. Kozlíková, H-Y. Wu, G. Johnson, D. S. Goodsell, A. Olson, E. Gröller and I. Viola, "Labels on Levels: Labeling of Multi-Scale Multi-Instance and Crowded 3D Biological Environments", in *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 977-986, Jan. 2019, doi: 10.1109/TVCG.2018.2864491.

Paper B: D. Kouřil, T. Isenberg, B. Kozlíková, M. Meyer, E. Gröller and I. Viola, "HyperLabels: Browsing of Dense and Hierarchical Molecular 3D Models", in *IEEE Transactions on Visualization and Computer Graphics*, 2020, Early Access, doi: 10.1109/TVCG.2020.2975583.

Furthermore, Chapter 6 is based on a finished work that is currently being prepared for peer review:

Paper C: D. Kouřil, O. Strnad, P. Mindek, S. Halladjian, T. Isenberg, E. Gröller and I. Viola, "Molecumentary: Scalable Narrated Documentaries Using Molecular Visualization", Nov. 2020, preprint, arXiv:2011.02418 [cs.HC].

Besides these three first-authored research works, the author has contributed to these related publications:

S. Halladjian, H. Miao, D. Kouřil, M. E. Gröller, I. Viola, T. Isenberg, "ScaleTrotter: Illustrative Visual Travels Across Negative Scales", in *IEEE Transactions on Visualization and Computer Graphics*, 26(1):654-664, January 2020. Doi: 10.1109/TVCG.2019.2934334

H. Miao, T. Klein, D. Kouřil, P. Mindek, K. Schatz, M. E. Gröller, B. Kozlíková, T. Isenberg, I. Viola, "Multiscale Molecular Visualization", in *Journal of Molecular Biology*, 431(6):1049-1070, March 2019. Doi: 10.1016/j.jmb.2018.09.004

T. Koch, D. Kouřil, T. Klein, P. Mindek, I. Viola, "Semantic Screen-Space Occlusion for Multiscale Molecular Visualization", in *Eurographics Workshop on Visual Computing for Biology and Medicine*, 197-201, September 2018. Doi: 10.2312/vcbm.20181245

P. Mindek, D. Kouřil, J. Sorger, D. Toloudis, B. Lyons, G. Johnson, M. E. Gröller, I. Viola, "Visualization Multi-Pipeline for Communicating Biology", in *IEEE Transactions on Visualization and Computer Graphics*, 24(1):883-892, January 2018. Doi: 10.1109/TVCG.2017.2744518

The rest of the thesis is structured as follows. We outline the state of the art upon which the work presented in the thesis builds in Chapter 2. Chapter 3 discusses the overarching

theme of navigating complex virtual environments and sets it in the context of the broader research field of data visualization. Chapter 4 describes a method for textual annotation of complex molecular models. Chapter 5 presents an interaction technique utilizing the textual labels, which serves for intuitive navigation of the same complex environment. Chapter 6 talks about automatizing the creation of scientific movies with real-time molecular visualization and text-to-speech technology. Finally, in Chapter 7, we conclude by discussing the research presented in the thesis and envision possible future research directions.

1.3 Authorship Statement

This thesis' content comprises research works done by its author, David Kouřil, in his doctoral studies. However, the individual research projects described in Chapters 4, 5, and 6 are the results of interdisciplinary and collaborative efforts[1]. In this section, we detail the contributions of the co-authors of the corresponding manuscripts.

- **Paper A** (Labels on Levels)

  The first project was motivated by a previous collaboration between The Scripps and TU Wien. The two co-authors from The Scripps—David Goodsell and Arthur Olson—provided the biology domain background and helped evaluate the technique. Graham Johnson was a third biology domain person who contributed with his evaluation as an experienced biological illustrator. Ivan Viola formulated the initial vision for the project and supervised the research. The method's design has been mostly established by me, Ivan Viola, and Eduard Gröller through frequent in-person meetings. The technical realization of the proposed method has been done mostly by me. I was also responsible for the paper writing with the help of Ladislav Čmolík, who also contributed with his previous knowledge about label layout generation methods. The paper writing was further aided by contributions from Hsiang-Yun Wu (label placement expertise) and Barbora Kozlíková (molecular visualization expertise). Eduard Gröller and Ivan Viola ensured the quality of the final manuscript and contributed the final edits.

- **Paper B** (HyperLabels)

  This work was again triggered by a conceptual idea from Ivan Viola. I was responsible for the detailed design of the navigational technique, algorithmic development, and its prototypical implementation. The method was finalized through an iterative cycle, incorporating high-level feedback from all the other co-authors. I drafted the manuscript story and wrote most of the text. Tobias Isenberg contributed to the final paper exposition. Barbora Kozlíková further polished the paper's exposition. Miriah Meyer contributed with ideas for visualizing hierarchical data and shaped

[1] In this thesis, the pronoun "we" is used to reflect this collaborative work.

the paper structure. The final manuscript was reviewed and edited by Eduard Gröller and Ivan Viola.

- **Paper C** (Molecumentary)

  The third project idea was developed in meetings between me and Ivan Viola, who supplied the initial vision. Together we devised a research plan based on a state-of-the-art review and defined the components of the technique. I drove the design of the resulting method with input from the other co-authors. The whole project underwent iterative development with frequent online meetings during the ongoing global pandemic. The technical implementation was carried out by me with the help of Ondřej Strnad, who mainly worked on the speech generation part of the technique. I outlined the narrative of the paper manuscript and wrote most of its parts. Peter Mindek contributed with his expertise in procedural animation and helped with the drafting of the paper. Sarkis Halladjian further contributed to the writing of the manuscript. Tobias Isenberg, Eduard Gröller and Ivan Viola steered the research and helped contextualize the proposed method in the data visualization field. The last three co-authors were also responsible for the final editing of the submitted manuscript.

CHAPTER 2

State of the Art

In this thesis, we visualize molecular models constructed by structural biologists. From the visualization perspective, the structural models feature a unique combination of characteristics that ought to be considered in visualization and interaction design. This chapter reviews the visualization literature and highlights related works that we can leverage to fulfill the research goals specified in Chapter 1.

2.1 Data Characteristics

We work with *mesoscale* molecular models in this thesis. In an effort to bridge the gap between the nano-scale of molecular biology and the micro-scale of cellular biology, scientists have created methods to integrating several data sources and generate models at the *meso*, i. e., intermediate, scale [GFH18]. Most of the models used in the thesis were created by the cellPACK [JAAA+15] tool. cellPACK uses multiple data sources integrated into a molecular model description, a so-called recipe. The recipe lists ingredients, i. e., molecular components, and their concentrations and distributions based on the current understanding of the modeled organism's composition. The user also provides a surface description of the compartments of the wanted model. A procedural packing algorithm then generates a 3D model using the ingredient components based on the recipe description. The result is a three-dimensional model that cannot be attained either by the cellular level imaging techniques (i. e., electron microscopy) or by crystallography used to determine the molecular level due to the limited range and resolution.

The resulting biological model posses characteristics that are unique for scientific visualization. We consider the following characteristic aspects of a biological model in the design of the navigation. First, the models are **multi-scale**. They contain interesting or important structures of varying sizes, from the level of a single atom (approx. 0.1 nm) to the level of a cell (approx. 10 µm). These levels are realized by a **hierarchical** organization, i. e.,

smaller structures such as molecules bind into larger molecular complexes that contribute to the build-up of even larger biological structures. Furthermore, each type of structure can be, and in most cases is, represented in the environment by very many instances, making the model **multi-instance**. The incredible amount of structural instances then results in a very **dense** packing of objects in the 3D space. These characteristics are significant for two reasons. First, they bring additional complexity that must be addressed in navigating these environments. Second, they themselves represent aspects of the data that should be communicated to the visualization user.

2.2 Molecular Rendering

Molecular graphics, i. e., using computer graphics to represent molecular structures, have been historically interlinked with the visualization field. It provided one of the first use cases for scientific visualization [Fra02]. The initial challenge of rendering even a single small molecule has, through many years, led to the development of many visual tools for various analytical tasks, such as computation of tunnels in protein structures for novel drug design [KKL+16]. As the field developed and the technology improved, visualization experts have become able to depict more extensive complexes.

Over the years, many visualization tools have developed into full-fledged suites today used by scientists in their everyday work. Among the most popular belong VMD [HDS+96], Chimera [PGH+04], and PyMOL [DeL02]. Besides the tools developed in the biology domain, several tools have matured over the years through visualization researchers' efforts. For example, MegaMol [GKM+15] is an open-source framework originating in the visualization community. It implements high-performance GPU methods to handle large particle datasets in real-time. For a broader overview of the molecular visualization field, together with an extensive review of available toolsets, we refer to Kozlíková et al.'s survey [KKF+17].

As structures on the molecular level do not possess a specific visual appearance, due to their sizes smaller than the wavelength of visible light, many different representations have been proposed for their visual depiction in molecular graphics. For visualizing large molecular models, the van der Waals (vdW) [Ric77] surface representation, where each atom is depicted as a sphere, is most widely used. In the past decade, researchers have proposed techniques that leverage the GPU's parallel processing power and the programmable shader pipeline to render the vdW molecular surfaces with real-time framerates. Tarini et al. [TCM06] initiated this work by using 2D impostors to draw the spherical atoms. Daae Lampe et al. [DVRH07] improved this technique through the inclusion of geometry shaders. Le Muzic et al. [LMPSV14] further improved the imposter method's performance through their use of the tessellation shader stage.

Nowadays, researchers can interactively render molecular structures with tens of billions of atoms. Each of the billions of atoms projects to a size smaller than a single pixel at this scale. Level of detail strategies have been proposed to optimize the rendering process and help with comprehension of the multiple levels of a molecular model's

organization [LBH12, FKE13]. Reducing visual clutter through illustrative methods has led to an investigation of visual abstraction. Early work by van der Zwaan [vdZLBI11] showed smooth transitions between rendering styles—and levels of abstraction—on molecular structures. Later, Viola and Isenberg [VI18] explored abstraction in the broader context with the aim of defining this previously loosely-delineated term in the visualization field.

Artistic depictions inspire many illustrative and abstracted rendering styles. David Goodsell [Gooa, Goo09] is a prominent scientific illustrator who defined the visual style of mesoscale structures. The work in this thesis is based on the visualization software described by Le Muzic et al. [LMAPV15], which uses Goodsell's illustrations as their primary inspiration. Their tool, called cellVIEW, implements a rendering pipeline that leverages the full programmable shader pipeline to visualize mesoscale structures in an illustrative manner at high framerates. For the research described in this thesis, their rendering method was re-implemented in a framework designed for biology communication called Marion [MKS+18].

Visualizing large molecular datasets at interactive framerates naturally leads to questions related to interacting with such visualization. Next, we review work in three research areas of interactive visualization. First, we look at navigating 3D environments, which, in most cases, translates to manipulating a virtual camera. We also focus our review on techniques that deal with scenes featuring multiple scales. Second, we explore occlusion management techniques. Dealing with occlusion is inherently needed in the navigation of dense molecular models. Finally, we review methods for annotating visualizations with textual labels. Labels are a prominent topic in this thesis, as they serve several functions and contribute to navigation and comprehension of complex molecular environments.

2.3 Navigation in 3D Multi-Scale Environments

Several authors discuss aspects of 3D navigation. Jankowski and Hachet [JH13], for example, surveyed many interaction techniques for navigating 3D environments. In addition, Christie et al. [CO09] 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 user's task goals. Both surveys focus on macro-space techniques and ignore multi-scale settings.

Early systems like Pad [PF93] and Pad++ [BH94] can be considered as the first multi-scale environments in computing. Furnas and Bederson [FB95] formalized the concept of 2D multi-scale environments with their Space-Scale Diagrams. Mackinlay et al. [MCR90] 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 quickly skip through the uninteresting (empty) space, while facilitating fine navigation when inspecting a target from a close distance. Ware and Fleet [WF97] extended this concept by changing the camera speed based on a few samples

from the current depth buffer. McCrae et al. [MMGK09] 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. [TRCC01] improved users' performance 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. Again, the technique is tailored for a macroscopic world, assuming that all objects reside on a ground surface.

Multi-scale navigation is particularly important for visualizations where the data covers multiple orders of magnitude, e. g., in astronomy. OpenSpace [BAB+17], for example, relies on the Dynamic Scene Graph [ACS+17] to solve the problem of limited floating-point number precision for positions of individual objects. Sagristà et al. [SJMS19] 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 sufficient precision for high dynamic distance ranges. A recent example of nano-scale visualization is Halladjian et al.'s ScaleTrotter [HMK+20] 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 [HMK+20]. Individual planets or stars are separated by a vast amount of empty space, whereas a cell's environment is crowded by omnipresent and always-moving molecules, which we address with our approach.

Approaches for *automatic* camera control have been proposed for several applications. For multi-scale environments, Van Wijk and Nuij [VWN03] presented a theoretical framework for 2D map navigation with smooth zooming and panning transitions. Their concept has been used in many fields and has also been extended to 3D settings [AE05].

Christie et al. [CON08] presented a detailed overview of the problem to control a camera in virtual 3D environments. According to them, automated camera control is needed in common computer graphics applications and, more specifically for our work, in multimodal systems. In the particular case of graphics and language (text or speech), the linguistic reference to an object dictates that the latter is not occluded, for example by using cutaways and ghosting [SF93]. In addition, spatial prepositions (e. g., in front of, left of) and dimensional adjectives (e. g., big, wide) add constraints to the camera.

The path of a moving (virtual) camera is also an important aspect of camera control. Collision avoidance in complex 3D environments, visibility of multiple targets, and smoothness of the trajectory are all challenges for the path computation [CON08]. Salomon et al. [SGLM03] used path planning for interactive navigation in complex

3D environments. To achieve this goal, Oskam et al. [OSTG09] constructed a visibility-aware roadmap graph and pre-compute an estimation of the pair-wise visibility between all elements of the environment. Then, their algorithm traverses the graph and computes large, collision-free transitions in real-time. Path planning techniques have also been proposed for digital cameras, particularly those attached to drones. Galvane et al. [GLC+18] proposed a cinematographic path planning technique. The authors relied on a visibility-aware roadmap, similar to Oskam et al. [OSTG09]. They generated a qualitative path (in terms of cinematographic properties) by finding the shortest path in Toric space instead of world space. Nägeli et al. [NMD+17] applied local avoidance techniques to follow pre-designed camera paths. Both of these techniques take dynamic targets and obstacles into account. Knöbelreiter et al. [KBUF14] also proposed an automatic generation of flythroughs for architectural repositories. Finally, Christie et al. [CMN+05] provided a survey of virtual camera planning methods.

In addition to viewpoint placement and camera path planning, cinematography addresses other issues such as shot composition, lighting design, staging (actor and scene element positioning), and editor requirements [CON08]. Amini et al. [AHRL+15] analyzed professionally designed data videos and cinematography and data design workshops. They extracted principles useful in designing comprehensible data videos. Burtnyk et al. [BKF+02] proposed *StyleCam*—a system of camera control integrated into a 3D model viewer. Their system was designed to create highly stylized animations of 3D scenes, such as commercials or feature films. In their follow-up work, Burtnyk et al. [BKFK06] introduced a method for presenting 3D models by using a dynamic camera. Their system *SlowMotion* used various cinematic transitions to maximize the presented model's visual appeal. Mindek et al. [MvV+15] developed a method for summarizing multiplayer video games that merges views from the participating players (i. e., a *flock of cameras*) into a single coherent movie, visually narrating the gameplay. Lino et al. [LCL+10] proposed a fully automated system that produces movies of 3D environments in real-time, with a focus on cinematographic expressiveness. By taking the 3D environment and narrative elements as input, their system computed Director Volumes with multiple associated cinematographic properties (e. g., visibility, camera angle, shot size). Then, the system edits the Director Volumes by enforcing continuity rules and computing transitions. In our work, we use several concepts from automatic camera control and cinematography to maximize the information value of our generated visual narratives. We mainly focus on navigation within dense environments of multi-scale molecular models, where not all of the common practices can be efficiently applied.

2.4 Occlusion Management in Dense Models

Due to the dense packing, we also need to adjust the scene's visibility settings to prevent occluding objects of interest. Elmqvist and Tsigas [ET08] 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 several objects' spatial arrangement—is most relevant in the thesis.

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 [LACS08, BG06], deformations [CSC07, CSC06, MTB03], or displacements [Elm05, EET07]. Viola and Gröller [VG05] summarize occlusion-handling approaches in visualization, which address the issue in a *smart* way, i. e., leveraging the underlying features of the model or domain as well as the user's expertise. We leverage the hierarchical and encapsulating character of meso-scale molecular models to create view-dependent cutaways in our work automatically.

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 [TFBB14] or depicting possible binding locations for protein interactions [FBG+18]. Le Muzic et al. [LMMS+16] 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. [SMR+17] 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 [FS92], through the definition of rules for automatic cutaways generation [DWE03], to techniques that give users more control of the shape of the cut [LCKM05, CH06]. These solutions emphasize generating illustrations, yet the demonstrated cases use relatively simple models compared to our molecular models from biology.

Particularly interesting is the work by Li et al. [LRA+07], 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 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.

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 specific 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 [BGKG06, KSW06, BHW+07].

Viola et al. [VKG04] formalized how less important structures are suppressed to prevent occluding more salient structures in the model. This concept has later also been applied to polygonal data [BF08]. Building on the importance-driven rendering, Viola et al. [VFSG06] 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.

2.5 Textual Annotation in Scientific Visualization

Methods for positioning labels, i.e., short textual annotations, in two- and three-dimensional scenes have been intensely investigated for decades. The general goal of labeling methods is to algorithmically create aesthetic and compact label layouts where all labels are readable and unambiguously associated with the labeled objects. The problem of finding the optimal distribution of labels, known as the label layout generation problem, has been proven to be NP-complete [MS91]. As a result, a number of approximation methods have been proposed over the years. We review two categories of works related to the thesis. First, we outline techniques for labeling 3D scenes. Second, we review methods that work with multiple scales, which has previously been the domain of geovisualization applications.

2.5.1 Labeling of 3D scenes

Labeling of 3D scenes, i.e., positioning of textual annotations in 3D space, is utilized mainly in the medical domain [OJP14] where both the spatial relations between the objects and their semantic description is needed to understand complex structures and processes in a human body.

In general, labeling approaches typically operate in the screen space on the projected image of the 3D scene. We distinguish between two types of labels. *External labels* are placed in the free space around the projected 3D objects and are connected with the projected 3D objects through leader lines. *Internal labels* are placed directly over the projected 3D objects or are just touching them.

We organize the external labeling approaches according to their strategies to localize the free space around the projected 3D objects. Preim et al. [PRS+95] and Huang et al. [HPL14] enclose the projection of the 3D scene with an axis-aligned rectangle

and position the labels in the free space outside this rectangle. Ali et al. [AHS05] and Hartmann et al. [HAS04] enclose the projection of the 3D scene with its convex hull and position the labels in the free space outside of the hull. Later, Čmolík and Bittner [ČmolíkB10] provided a GPU implementation of this approach. Positioning the labels around the convex hull instead of the bounding rectangle results in a more compact label layout. An even better result can be produced if the labels are positioned in all the free space around the projected 3D objects. Stein and Décoret [SD08] presented a greedy algorithm that positions the labels in the free space around the projected 3D objects. The algorithm evaluates whether the labels fit into the free space using a summed area table [HSC+05] and uses shadow regions to prevent the overlapping of labels.

The main limitation of all mentioned external labeling approaches is that they require free space around the labeled objects in the 3D scene or around proxy objects bounding the projection of the 3D scene. This precondition cannot be expected in very dense scenes, such as in our molecular biology environment. For such dense 3D scenes, internal labels placed over the projections of the 3D objects are more suitable. Bell et al. [BFH01] introduced the very first algorithm for internal labeling. They approximate each projected 3D object by the smallest axis-aligned enclosing rectangle. The approach determines the free space based on the rectangles. It places each label at the center of the largest rectangle contained in the projected 3D objects that is not occluded by any other rectangle. The size of the labels is determined by the free space available and predefined constraints. If a label cannot be placed internally, it is placed externally in the free space outside of the object and connected with the object by a leader line. Other approaches [GAHS05a, GAHS05b, GHS06a] directly use the projected 3D objects and their visibility to evaluate the free space for internal and external labels. The internal labels are positioned based on the skeletons of the projected 3D objects.

Internal labels are also adjusted according to the shape of the labeled objects. Mass and Döllner [MD06] use object-integrated billboards to position labels over rectangular 3D objects, such as buildings. Ropinski et al. [RPRH07] approximate the 3D objects with Bezier patches and align the labels with the geometry of the underlying structures. Cipriano and Gleicher [CG08] provide a solution for surface regions with high curvature, where the surface-aligned labels would be distorted. They create scaffold surfaces onto which the labels are placed.

Prado and Raposo [PR13] internally label 3D objects with spherical, cylindrical, and rectangular shapes in object space. The occlusion of the labels by the 3D objects is not considered, but the labels are positioned several times on each 3D object.

None of these techniques deal with a multi-scale situation, i.e., the presence of objects on different levels of detail and corresponding levels of labels. To the best of our knowledge, the only 3D scene labeling approach that considers multiple scales was presented by Götzelmann et al. [GHS06b]. The approach takes into account only two scales. The labels are organized into several groups and in each group they are positioned close together outside of the convex hull of the projected 3D scene. Possible crossings of the leader lines between different groups are not resolved.

2.5.2 Scale-Aware Labeling in Cartography

Although placing name labels is essential in cartography and has been extensively investigated in geovisualization [Imh75, SMKH08, Wol09, CLK14], labeling multiple objects in multi-scale environments is studied separately. Online map services usually place anchors to emphasize objects of the same type along the different zoom scales since the objects' positions are predefined and can be easily computed. Lin et al. [LKY08] introduced *Many-to-One Boundary Labeling*, where they placed exactly one external label for objects of the same type along the boundary of the map domain. This has been extended to study placements without crossings [Lin10] and along specified backbones [BCF+15], but no scale-dependent approach has been proposed.

Initially, researchers studied static map-labeling approaches, which are insufficient for navigation purposes due to their high computational requirements [BGS+13]. To solve this, fast dynamic labeling techniques were developed for moving and zooming in/out of the map domain. Acceleration is achieved by decomposing the map content into several scales and only the significant objects will be annotated [BSB+12]. Nonetheless, scales were mostly handled independently and thus visual consistency was often ignored [FP03, Mot07]. Poon and Shin [PS05] developed the pioneering approach to build a hierarchy tree to guide label placement for user navigation. Been et al. [BDY06, BNPW10] then extended the shape and orientation of labels to develop active range optimization (ARO) for general dynamic labeling purposes. The idea is to maximize the persistence of name labels by computing placement conflicts using rectangular pyramids in the zoom space. Zhang et al. [ZPLC15] extended pyramids to trapezoidal boxes to constrain lower and upper bounds of labels along zoom scales. Wu et al. [WTPA17b, WTPA17a] introduced the hierarchical structures of datasets into the label placement process. This is done together with optimizing the label movement and the leader lines to improve labels at each scale. Other researches focused on slider-based [SHWZ14] and 2D rotation-aware [GNR16] label placement, while artifacts usually occurred during exploration. Slider-based labeling uses labels that slide along their anchor points. Rotation-aware labeling approaches determine an optimal label placement that is consistent upon a 2D rotation of the map.

Unfortunately, these scale-aware techniques cannot be easily adapted to the labeling of 3D scenes because they do not consider camera rotation with three rotational degrees of freedom. Further, there is no unambiguous definition of the level-of-zoom in a 3D environment for a whole rendered frame. In contrast to scale-aware labeling of interactive maps where the level-of-zoom is defined from the sea level, we cannot define the level-of-zoom as a variable with one degree of freedom for 3D scenes as the distances of 3D objects from the camera are not constant.

2.5.3 Labeling in the Visualization of Biological Environments

Labels are widely used in molecular and mesoscale visualizations, but with significant limitations. Many effective molecular graphics programs are available, such as Jmol/JSmol [Her06], Chimera [PGH+04], and PyMol [Sch15]. They are essential tools

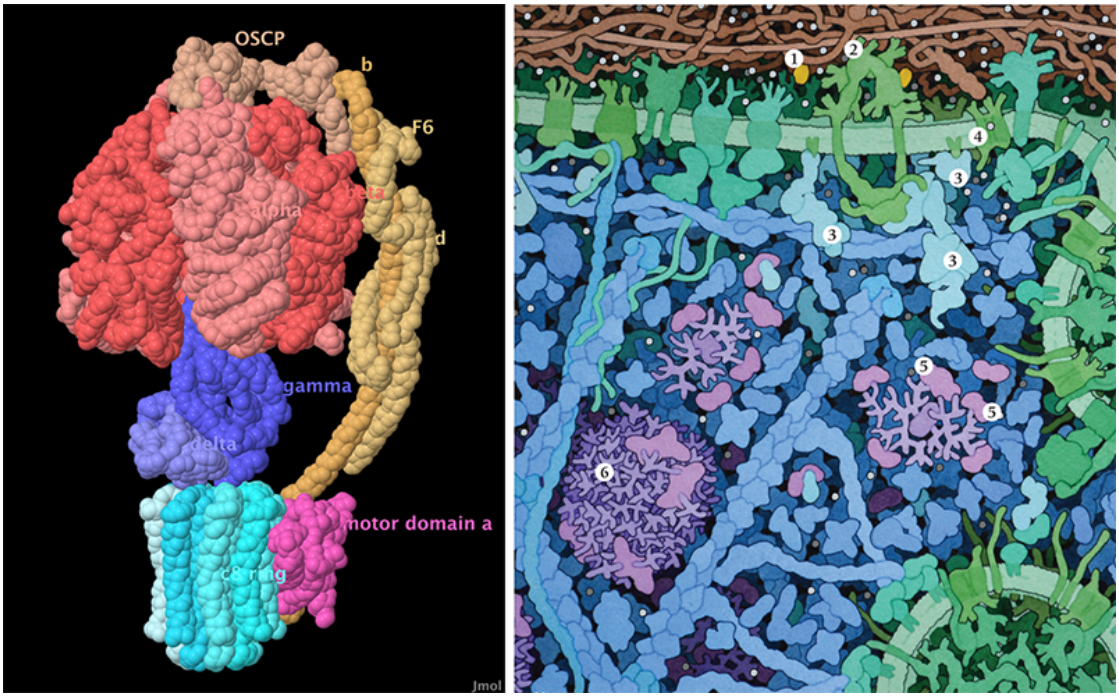


Figure 2.1: Left: JSmol image of ATP synthase, with subunits labeled. Atoms were chosen manually for the placement of each label. With the default settings, labels are occluded by the spherical atoms, so several labels are not visible in this view. Labels are colored based on the colors of the subunits. Right: Detail of a poster presented at the educational portal of the RCSB Protein Data Bank, showing molecular processes in insulin signaling. An accompanying caption is used to identify each molecule (1-insulin, 2-insulin receptor, 3-signaling proteins, 4-glucose transporter, 5-glycogen biosynthetic enzymes, 6-glycogen).

in structural biology research and outreach. Two types of labeling are implemented in these tools.

First, information about individual atoms, including atom names and corresponding amino acid and polypeptide chain, may be obtained by hovering over an atom in the rendered frame. In our experience, this is indispensable in day-to-day research if combined with flexible scripting and coloring tools.

Second, 3D labels may be attached to user-defined atom positions. A variety of customizable parameters, e.g., color, font size, offsets, shaded backgrounds, and different approaches to occlusion, allow users to optimize the frame. Problems with overlap and visibility continue to plague these types of labels in all but the most simple applications. Figure 2.1 (left) shows an example of labeling in JSmol. In this case, considerable time is required to manually pick appropriate atoms for placing labels and to reduce occlusion problems while interactively manipulating the object.

There is also a long history of labeling 2D atomic, molecular, and cellular structures, such as those used in textbooks and professional publications. These examples may be used as style guides when designing interactive labeling approaches. The labels are typically added to the imagery in a post process, allowing the designer to optimize the placement and other characteristics. Effective designs often include approaches like drop shadows or outlines to provide contrast between labels and the image. Furthermore they use blank space (if present) around the main object, and add line breaks or abbreviations to the text to reduce the horizontal extent of labels. Static images also provide the freedom to place most of the label information into an accompanying caption or legend, using simple reference numbers or letters to identify features in the image itself (see Figure 2.1 right). With static printed images, users are typically willing to devote more time for switching attention between figure and caption, in a way that would be prohibitive for interactive applications with continually changing views.

In addition, complex static images are often labeled with numbers or letters to keep occlusion low (see Figure 2.1 right). The label serves as reference to text near the illustration called a “key", which contains the label and often a more extensive description of the object. This is one of the typical labeling scenarios used especially in 2D static images and scenes. However, this solution is not feasible for dynamic and complex 3D scenes as the user constantly needs to switch attention between observing the label and its description.

In the case of interactive environments, we can afford to place the entire labels directly into the scene, close to the position of the corresponding object. This also fits the recommendation of Tufte [Tuf01] to place labels directly on the graphic itself, without legends, and close to the labeled features. This avoids using the leader lines between the features and their labels. The user can temporarily turn off the labels and examine the structure itself, which solves possible occlusion problems.

2.5.4 Clickable Labels in Visualization

While the labels are mere carriers of information in the techniques mentioned so far, other approaches facilitate interaction by clicking on a label. Clickable labels are especially prominent in visualizations for anatomy education [PS18]. As an early example, ZoomIllustrator by Preim et al. [PRS97] 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 [MP09] 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 [JD12] 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.

Interactive labels can also serve for orientation in a hierarchical structure. Lerin et al. [LYT12] 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 continuous animation. Similar hierarchical interaction element can be found also in Kopper et al.'s work [KNBP06]. 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.

Clickable labels play a vital role in this thesis. They serve the function of annotation that is essential in communicating phenomena depicted in a visualization. The integration of interactivity furthermore facilitates navigation. The interaction scheme of clicking—or touching—on a textual link is very well understood by the general public and therefore provides an intuitive interface even for lay-person users. For that reason, labels represent a suitable platform for science communication in the interactive molecular visualization environment. In the following section, we overview methods used for communicating scientific facts through visual means. Furthermore, we outline storytelling research in the visualization context and argue for coupling the audio channel with the visuals for explanation purposes.

2.6 Science Communication

In general, biology experts very well understand the nature of the complex and unobservable environment of large molecular models. They spend their professional lives thinking about the processes happening on these scales. However, it is tough for lay-persons to imagine how such an environment looks, which is where visualization can often help. That is why more and more focus has been placed on so-called *science outreach* in the past years. The goal of this effort is to bring scientific findings closer to the general public. There are many benefits to this endeavor [Var14]. With the commoditization of high-performance graphics hardware, the advances in computer graphics have reached natural sciences and found applications in communicating its research results. Not only is computer graphics now utilized in entertainment, i. e., animated movies, but the accessibility of modeling and rendering software has allowed scientists to use it for illustrations and animations depicting scientific concepts. In that case, it no longer serves for entertainment, but also education. The term *edutainment* is sometimes used as it provides educative value in addition to often being entertaining. A new job title—that of a scientific animator—has emerged [Iwa14].

Besides the usual computer animation pipeline that utilizes "offline" rendering methods, i. e., ray-casting or ray-tracing, techniques for generating moving images in *real-time* have been developing alongside. Games are probably the most prominent case of real-time graphics. Data visualization is also, for the most part, based on real-time rendering.

It brings the possibility of a user to interact with the imagery. Interaction is a crucial component for most information visualization systems, where the graphical elements are typically distributed in a 2D space. The usual workflow of data visualization used for research tasks typically consists of gathering data from various sources, mapping the data items to visual representations, and then using the visualization system to *explore* this data in search of a new discovery. This is a distinctly different end goal than using visualization to prepare content with the purpose of *explaining* a particular known scientific fact. However, with the advances of real-time graphics, interactive data visualization has matured to the point where the same system used previously only for research tasks can also be used for presentation purposes. Ynnerman et al. [YLT18] coin the term *exploranation* to describe this paradigm shift.

2.6.1 Visualization Storytelling

One of the visualization's main purposes is to transform information—which can be extracted from data—into knowledge. Storytelling can play a key role in helping viewers to absorb this knowledge and make visualizations intelligible to a variety of audiences [KM13, GP01, MLF+12, SH10]. Lee et al. [LRIC15] discuss how storytelling should be scoped within the visualization context, and visualization research has often relied on storytelling in the past [TRB+18].

Storytelling and visualization are thus tightly related. Storytelling communicates the visualization designer's intentions to the viewer, and visualization as a technique is a medium that allows storytelling to achieve its communication goals. Narrative visualizations combine conventions of communicative and exploratory information visualization to convey an intended story. Hullman and Diakopoulos [HD11] discussed different types of rhetorical techniques in narrative visualizations. Gratzl et al. [GLG+16] presented a method for authoring visualization-based stories by integrating the stages of data exploration, story creation, and presentation. Hence, the story can be produced from the provenance information of the exploration performed by an expert user. Kwon et al. [KSJ+14] proposed a web-based method for an intuitive enhancement of articles with visualizations by linking them to relevant text segments. Machine learning has also been employed for generating textual captions from charts [LXH+20]. Ren et al. [RBL+17] analyzed the design space of annotations used in information visualization and developed a tool for augmenting charts with annotations. According to their analysis, an annotation in the context of visual data-driven storytelling is characterized by two design dimensions: form and target. An annotation's purpose cross-cuts these two dimensions. In our work, text annotations (form) next to biological data items (target) enable the communication.

Storytelling methods have been applied in various areas of scientific visualization as well [MLF+12, AWM10]. Wohlfart and Hauser [WH07] proposed a method for story-driven presentation of volumetric datasets. Hsu et al. [HMC11] created multi-scale overviews of various datasets in a single image. Thöny et al. [TSS+18] discussed various storytelling concepts, providing an overview of the design and requirements for interactive storytelling within the area of 3D geographic visualization. Lidal et al. [LHV12] described

a sketch-based storytelling technique for capturing geological interpretations during terrain-exploration processes. Sorger et al. [SMR+17] proposed *Metamorphers*—a set of operators for authoring visual stories form molecular datasets by transforming them into comprehensible animations. Contrary to other storytelling techniques applied in scientific visualization, they took the hierarchical organization and the crowdedness of molecular datasets into account. We also focus on visualizing datasets exhibiting specific characteristics, e. g., with limited visibility and overall complexity.

Various methods have also been developed for explaining complex datasets, with possible applications in education. Liao et al. [LHM14] visualized volume datasets by creating animations. Their semi-automatic method eases the creation of such animations by analyzing the user's interactions during an exploration of the dataset. Vázquez et al. [VGHN08] linked text descriptions from an electronic anatomy textbook with annotated images of 3D models to support the teaching of the anatomy. Molecumentary builds on these two approaches by producing animations enhanced by text descriptions.

2.6.2 Audio in Visualization

Verbal expression is inseparably related to storytelling. In visualization, however, the audio channel is mostly not utilized. Naturally, there are differences in processing visual and auditory inputs, as discussed by Munzner [Mun14]. For example, audio is inherently linear and cannot be processed at once. On the other hand, the human visual system's ability to gain an overview at one glance has served as a fundamental property and a strong argument for representing data visually. However, we believe that supplying audio together with the visualization promises great potential for storytelling purposes. The dual coding theory implies that humans process visual and verbal inputs by separate systems [CP91]. This fact suggests potential in utilizing the verbal processing pipeline in communication and education along with visual comprehension.

Visualization can be enhanced with audio in two ways. Sonification refers to the use of non-speech audio to support visualization in communicating the intended message. Various data sonification toolkits and frameworks have been proposed, such as *Porsonify* [MR95] or *Listen* [WL96]. The second way is to apply voice-overs. They could be either performed by voice actors and recorded, or synthesized. There are various surveys on the vast number of works related to text-to-speech synthesis [Kar18, SVB17].

CHAPTER 3

Overview

Computing technology allows us to display environments that mirror those from real life in fidelity and scale like never before. We can render the whole observable universe with planets, planetary systems, stars, black holes, and galaxies. We can render the insides of a cell, theoretically, down to the atomistic detail. In most cases, we can even achieve real-time performance, i. e., rendering with interactive framerates in the 30 to 90 FPS range. Since these environments are now starting to find their way into the hands of even ordinary users, who can interact with the data and navigate across a magnitude of space and scale, the question of how to most efficiently navigate in such a *multi-scale* environment arises.

LaViola et al.'s book [LKM+17] comprises research efforts in navigating virtual environments across several modalities, from a conventional desktop to a highly-specific virtual reality hardware set-ups. They divide navigation into two parts: travel and wayfinding. While *travel* refers to the physical act of moving through an environment, the *wayfinding* term corresponds to the cognitive aspects of such activity—understanding the environment, establishing targets, and finding paths to reach them. In practical applications both travel and wayfinding processes are combined in the individual interaction methods, depending on the specific tasks. Also in our work we contribute methods that facilitate both the act of traversing of a large virtual space, and comprehension of the environment and its hierarchical organization across several magnitudes of scale.

Multi-scale virtual environments have an origin in 2D information spaces. Pad [PF93] and Pad++ [BH94] (Figure 3.1, left) are often cited as the initial example of an environment that represented several levels of organization. It borrowed from spatial thinking of cartography and organized data items on an infinite 2D plane. Two important concepts that, in essence, implement multi-scale navigation in the Pad systems are semantic zooming and portals. Researchers over the years thoroughly investigated multi-scale navigation in 2D space. Furnas and Bederson formalized the concept of multiple scales in 2D visualization through their Space-Scale diagrams. [FB95]. Later, van Wijk and

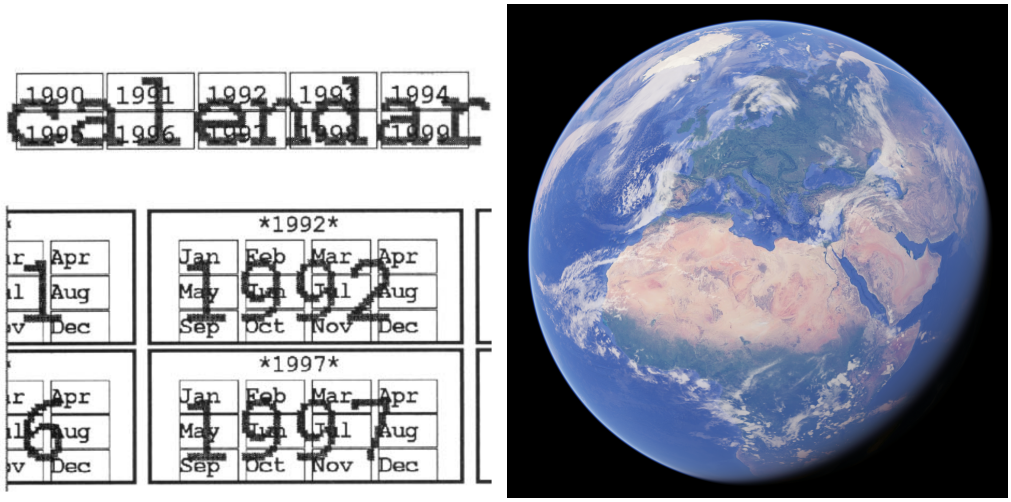


Figure 3.1: Two examples of 2D multi-scale environments: the Pad system [PF93] allowing to navigate a calendar, and Google Earth application that enables intuitive navigation on the surface of the planet Earth.

Nuij [VWN03] provided an analysis of panning and zooming transitions to smoothly animate between two areas of interest.

Geovisualization represents a natural use case of multi-scale visualization, especially when extended to the scale of visualizing the whole planet Earth surface. Such applications, also known as *virtual globe* or *digital earth* [MAAS15], often provide an intuitive navigation across the many zoom-levels. As an example, one can think of the Google Earth program (Figure 3.1, right). Even though the globe represents a three-dimensional object, i. e., roughly a sphere, conceptually, the navigation boils down to the 2D case due to the projection of the sphere surface.

The Google Earth example leads to an interesting question: At what point does the 2D navigation become 3D? At the lowest level, i. e., individual streets, Google Earth features three-dimensional buildings that are often extruded from the photographed planet surface. We do not consider this an example of a truly three-dimensional multi-scale environment since the detail is only available at the lowest level. Therefore, we find that the problem of navigating multi-scale environments that are truly three-dimensional is more complex, and the methods developed for the 2D case can only be applied in a limited manner [Zha09].

We consider two examples of a 3D multi-scale environment, illustrated in Figure 3.2: cosmology and biology. These two fields have similar features in many aspects but also carry significant differences, the most prominent one being the difference in the density of packing in the environment. For the *positive* scales (10^n m), the individual objects are separated by large amounts of empty space. At the *negative* scales (10^{-n} m), the environment is packed with objects significantly more densely, which complicates

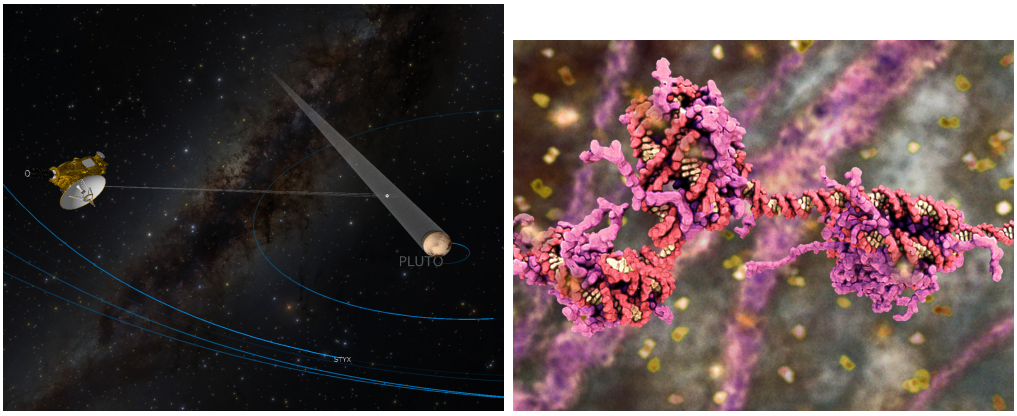


Figure 3.2: Two exemplary truly three-dimensional environments: visualizing a probe in the context of the observable universe using OpenSpace [BAE+18] (left), and an artistic depiction of nanoscopic biological structures by animator Drew Berry (right).

visualization. Figure 3.2 (right) depicts an artistic depiction of biological environment on these scales. The artist, scientific animator Drew Berry, specifically abstracts away most of the density to place focus on one specific structure.

Both cosmology and biology fields have benefited from the advances in real-time visualization. In visualizing the universe, scientists went from investigating the individual astronomical bodies (e. g., planets and stars) to being able to render datasets capturing the whole observable universe interactively. This shift to a massively multi-scale visualization is mirrored in biology. Similarly, researchers who previously examined individual molecules can now see it in the context of whole viruses, bacteria, or even cells. In this thesis, we focus on the case of visualizing biological structures. This environment comes with a unique combination of its characteristics that have implications for designing navigational methods. The *multi-scale* aspect means the user navigates not only in space but also in scale, i. e., traverses between structures of varying sizes. The *crowdedness* of the environment hinders the navigation—to explore a 3D model with dense packing by *very many* instances, it is required that some of the instances are cut-away. The question is how many instances is the navigational method allowed to cut-away to still maintain the impression of the model's crowdedness. Biological models are inherently *hierarchical* and the navigational method should ideally preserve and communicate this fact, too.

We mentioned before that science communication has previously been realized mostly in the form of pre-rendered animated movies. The utilization of real-time visualization in science communication brings the aspect of interactivity. A user's ability to interfere with the visualization raises the question of how the traditional way of communicating scientific facts relates to the new interactive medium. We envision these two approaches to create a whole *interaction spectrum* (Figure 3.3). On the left side of the spectrum, one may use a fully interactive real-time visualization system, with a potentially large number of parameters exposed. The user must almost act as a visualization expert to

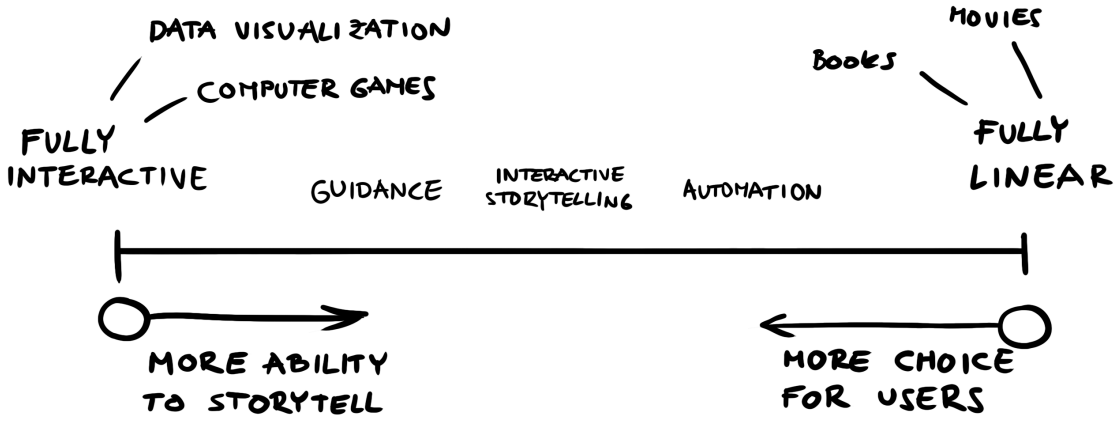


Figure 3.3: Illustration of the interaction spectrum concept: there are many options for communication using interactive methods between the two extrema. Methods closer to the left side allow more user intervention which can be an obstacle in telling a specific story. The other side of the spectrum supports telling of a story at the expense of user choice through interaction.

explore the visualization, and his or her participation is entirely active and manual. The other side of the spectrum represents approaches with authored content that follow a particular linear story structure. Such content has previously been implemented using a pre-rendered animation, but these days it can also be prepared using real-time graphics. In any case, the user becomes instead a passive viewer where their option of interaction is limited to, at most, controlling the playback.

However, the parameters that the user must tweak in the interactive scenario do not have to be either all controlled by the user or all pre-defined by the linear story. We will call these parameters "degrees of freedom" of the visualization. Degrees of freedom may be reduced through various means, e. g., automation, guidance, or other interactive storytelling methods. By reducing the degrees of freedom offered to the end-user, we fill out the space of possible visualization methodologies on the interaction spectrum. In our work, we focus on visualization parameters related to navigation and exploration of the 3D model. Therefore, we refer to them as *navigational degrees of freedom (NDoFs)*. We consider four principle NDoFs in the thesis—camera control, visibility control, annotation, and order of exploration—which we describe in the following section (Section 3.1). Afterward, in Section 3.2, we frame the individual research works of the thesis in a conceptual framework of *modes of navigation*. Each mode represents a general approach of interfacing users with a 3D visualization to navigate it. The modes also fill the interaction spectrum by presenting methods ranging from fully interactive to progressively more autonomous visualization systems.

3.1 Navigational Degrees of Freedom

Navigational Degrees of Freedom are components of a visualization system that influence navigation. In a conventional visualization system, each navigational DoF is under the user's control. He or she is responsible for adjusting its settings to get a meaningful and useful visual output. In principle, the fewer DoFs a system has, the more intuitive and straightforward to use it becomes. However, it also loses flexibility and the ability to adapt to an individual user's specific needs.

3.1.1 Camera Control

The first and the most apparent component needed for navigating a 3D model is a virtual camera used to project the 3D environment onto a 2D screen. Its placement and orientation both play a sizable role in the final frame, but other aspects like path planning come into play with animated transitions.

The direct way of controlling the camera is by mapping the mouse and keyboard interactions to a change of camera parameters. Typically, three basic operations are used: orbiting, panning, and zooming. These three tools are the standard across all major 3D authoring software, and 3D visualization applications often provide them as well. *Orbiting* refers to the rotating movement of the camera around a specific pivot point. The camera's orientation change is coupled with the positional change to keep the focus point constant. The pivot point can be changed by *panning*, which corresponds to a shift of the camera position parallel to the viewing plane. Finally, the *zooming* interaction moves the camera closer to the pivot point. In the 3D context, sometimes the term *dollying* is used as it better describes a virtual camera's movement towards the pivot point. Orbiting, zooming, and panning provide to the user a basic control over the camera. However, they also do not consider any semantical information of the scene to assist the navigation.

To stabilize the camera navigational DoF, we guide the camera movements or even fully automate the camera's path through the virtual scene. For example, Khan et al.'s HoverCam [KKS+05] maps the mouse interactions to smooth navigation around a 3D object, stripping the user from the need to orbit, pan, and zoom. Instead, the user simply hovers at the proximity of the object's surface. McCrae et al. [MMGK09] later used the HoverCam approach in combination with a cubemap of the camera's surroundings to implement multi-scale navigation. As an example of complete automation of camera movements, Burtnyk et al.'s *ShowMotion* system [BKFK06] generates cinematic camera movements to present parts of a model in a visually pleasing way. Viewpoint selection techniques [VFSH01] can also play a role in stabilizing the camera NDoF, as they enable to automatically select appropriate camera position and orientation to provide a good view of a 3D model's parts.

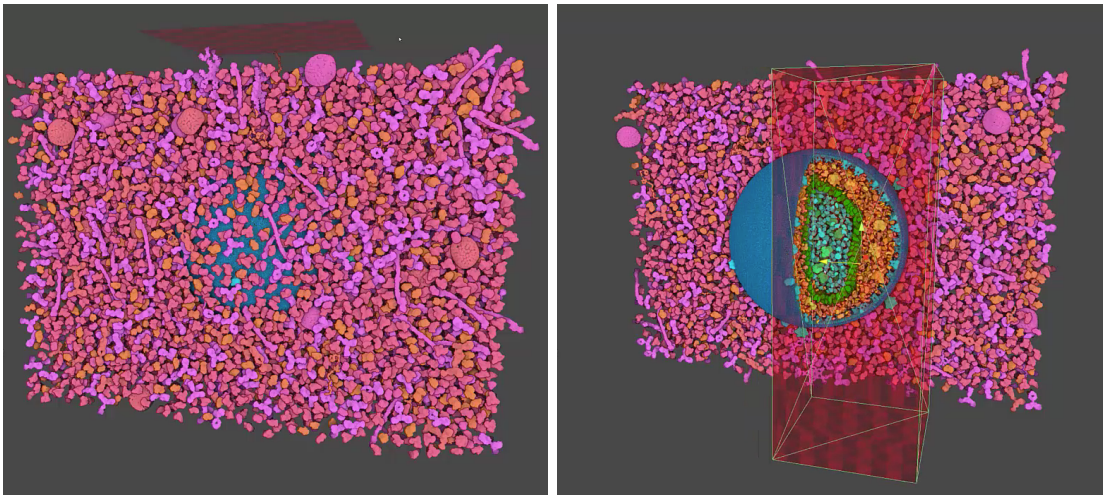


Figure 3.4: Le Muzic et al.'s Visibility Equalizer [LMMS+16] uses cut-objects to reveal the inside composition of a dense molecular model. In this example of direct control, the user has the possibility—and also responsibility—to fully control the model's visibility.

3.1.2 Visibility Control

The dense packing of biological environments requires to adjust the visibility of its parts to show and navigate such environment properly. Even after excluding all small molecules, e. g., water, the density leads to frequent occlusion. When showing all objects in a model, we provide an external view of the whole dataset and then show only the outer layer of objects. In case we position the camera inside the model, objects closest to the camera occlude the view of the rest of the model. To properly communicate the 3D biological model's interesting features, these two types of occlusion have to be addressed.

Again, direct control over the model's visibility settings can be entrusted to the user's hands. While there are many ways of handling visibility based on the model type (as described in Section 2.4), here we consider the simplest one, which is through cut-objects. Figure 3.4 shows an improved version of the basic cut-away objects technique implementation presented by Le Muzic et al. [LMMS+16]. It is explicitly designed for molecular models, allowing the user to tweak what structural types shall be affected by the cutting approach.

Nevertheless, controlling the model's visibility directly means that the user has to perform many actions to explore the model. The visibility control DoF can, therefore, be stabilized by automating some of the visibility parameter adjustments. Viola et al.'s work [VFSG06] represents an approach where a 3D model's visibility is resolved automatically based on a selected structure of interest. They present their solution on volumetric models. A similar technique for polygonal scenes was proposed by Li et al. [LRA+07].

3.1.3 Annotation

The third navigational degree of freedom we consider is annotation embedded in the 3D visualization. Any additional information integrated into the pure spatial data visualization belongs to this category. A 3D model with proper annotation can help users explore, for example, by guiding them to salient areas. Annotation can thus play an important role in navigation. As an example where such a navigational element is missing, we may imagine an interactive map that lacks annotation of cities, rivers, or mountains. Even though the visualization still shows the same spatial data, it is much harder to navigate and find specific targets. In such a scenario, one would have to look up external references and, for example, match the shapes of cities to confirm at which one they are currently looking.

Similar to a label-less map, a biological visualization without textual annotation lacks the proper semantics. When an untrained person sees a complex molecular scenery visualization, they cannot gain much information from just this image. The visuals are mostly incomprehensible to them. **Textual labels** are the first step of including more descriptive information about the meaning of the depicted structures. Labels bring in the element of annotation—they give names to what the user can see. They can also highlight important features that are at risk of being overlooked, e. g., small objects represented by a low number of instances.

Labels historically found extensive use in education. A notable example is the medical textbook Gray's Anatomy, which employs various textual labels to associate anatomical parts with their names in the book's illustrations. In this context, the illustrations are hand-crafted, and the labels are manually-assigned for the particular static image. The rise of computer graphics, especially real-time interactive visualization, has led to the development of methods for annotating real-time imagery.

Placing textual labels over scientific visualization essentially represents incorporating auxiliary information into spatial visualization. This additional information is often non-spatial, and it serves in various ways to augment the raw spatial data. In the case of textual labels, we incorporate *verbal* information to associate a 3D structure with its name. Additional non-spatial information integrated into a scientific visualization can contribute not only to annotation, i. e., assigning spatial structures with their names, but also for other purposes. We highlight two instances from our work: a) embedding a representation of the hierarchical organization for *orienting* the user, and b) inclusion of longer descriptions for *explanation* of the shown biological structures. In the first case, we try to prevent the user from getting lost in the large and complex scene. The probability of a lay-person user, who has limited domain knowledge, losing orientation in the unfamiliar-looking environments is reasonably high. We, therefore, include information about the current location in the hierarchical structure. More information about this type of information augmentation is in Chapter 5.

In the second case, we include more explanatory information to comment on the presented model. There is much more exciting information about biological models that would

ideally be passed on to the user. When an experienced biologist looks at a visualization of a virus, they can see hundreds of stories worth being told. For an uninitiated spectator, however, even elementary information can be fascinating. One way to accomplish the integration of more explanations is through embedding longer text descriptions, possibly synchronized with user interaction to provide appropriate commentary at specific times of the user's exploration. Larger text chunks, however, can occupy too much screen space. Furthermore, new knowledge comes in almost every day, and often, before a set of explaining texts is prepared, it becomes obsolete because of further discoveries. Another option is to supply the description through a voice-over. This is the most natural way of communicating interesting facts in a scientific context, exemplified by a long tradition of science documentaries. The same problem holds here, too: the time it takes to write the descriptional text and produce a voice-over recording might be longer than how long the fact stays valid. New knowledge resulting from ongoing research about the depicted phenomena triggers the whole production procedure again.

3.1.4 Order of Exploration

There is a final, somewhat abstract, degree of freedom available in every visualization system where interaction is enabled: the order in which the user explores the visualization. Through the interaction with the visualization, the user gains insight into the data.

Several high-level guidelines have been proposed in the visualization field that became the de-facto standard for visualization system design. The most prominent such principle is Shneiderman's visual information seeking mantra [Shn96]: "overview, zoom & filter, details-on-demand". Another common general concept is "focus+context". It encompasses the methodology of incorporating surrounding, contextual information for a zoomed-in, detailed view. While these principles mainly guide the design of visualization and the interactions offered in the visualization system, they also, in a way, express patterns for the exploration order.

Besides the general guidelines for visualization exploration, researchers have also investigated more specific elements of storytelling in the context of exploring a visualization. Segel and Heer [SH10] develop a design space of storytelling approaches in visualization. They propose seven narrative visualization genres based on an analysis of existing visual narratives. Ma et al. [MLF+12] suggest principles of good storytelling with visualizations of scientific data. They argue that in this case, the key story points stem from seeing data that are generally invisible. The ability to extend our senses, and glimpse into the workings of life, drives the storytelling in the scientific context. Wohlfart and Hauser [WH07] talk about dividing the control between the visualization story author and the user/observer. They recognize four possible modes: passive story telling, story telling with interactive approval, semi-interactive story telling, total separation from the story. Preim and Meuschke [PM20] survey extensively the use of animation for medical data, which, as they say, remains an underestimated way of communicating concepts from this field. Storytelling through interactive visualization continues to be an active

research domain. Kosara and Mackinlay [KM13] discuss the potential of storytelling as an important direction in visualization research.

Traditionally, we think of the manner and order of exploration as something that should be entirely in the user's hands. It is the reason we develop interactive visualizations. However, we have also seen how exploration can be improved by including aspects of storytelling or guidance. In such a case, the responsibility for finding interesting or important pieces of information is not entirely on the user's shoulders. We can think of alleviating him or her of this responsibility to an even greater degree. Automated control can be desired on some applications and venues, as described in the next section. By transferring the responsibility of deciding where to navigate in the 3D visualization, we stabilize this degree of freedom and free the users from having to decide themselves.

3.2 Modes of Navigation

In this section, we frame the contributions of this thesis in terms of a conceptual framework of *modes of navigation*. As we described in the previous section, in navigating a 3D scientific visualization, the user typically controls many parameters—what we call degrees of freedom. The baseline for the three stages described in more detail in the following text is so-called *free navigation*. Here we consider the case where the user is presented with purely the visualization of the spatial data and is given basic conventional tools for each of the degrees of freedom. Therefore, the user has full manual control over all the navigation parameters and has absolute command over the exploration of the data.

The navigational modes are defined on the interaction spectrum. One side of this spectrum represents the typical data visualization approach, where the system is fully interactive. The other extreme of the spectrum is dedicated to approaches that, on the other hand, do not allow any form of interaction to influence the visualization. We imagine an animated movie as an example of this side of the interaction spectrum.

In a way, this problem is related to *interactive storytelling*. The gap between traditional storytelling media, such as movies, and new media that allow user interaction has been debated since the turn of the century. Glassner [Gla01] discussed how stories and games (i. e., interactive graphical applications) are in some ways incompatible. For example, while stories are consumed passively, games by definition require active participation. Further differences between the two media are how they communicate, how actions are decided, and how rules are defined and enforced. Perlin [Per05] later discussed how characters and their interactions are the key aspects of, potentially interactive, narrative. In our case, we consider a particular sub-genre of storytelling—a scientific documentary—where the storyline communicates scientific findings.

Several researchers have previously addressed the idea of steering user exploration in large information spaces. Dennis and Healey [DH02] present a navigation assistant component that identifies areas of interest in a multidimensional weather visualization and designs animated transitions between these salient regions. Another way to look

at the interaction spectrum is that each navigational mode includes progressively more *guidance*. Guidance has been investigated for visual analytics systems by Ceneda et al. [CGM+17]. They define a design space of guidance for visual analytics and include the concept of a *degree of guidance*. They recognize the same two extremes: on one side lies a system with no guidance while a fully automated system represents the other extreme. They pinpoint three scenarios that lie on the space between the two extremes: *orienting*, *directing*, and *prescribing*. They focus on visual analytics systems and tasks which are different from our case of scientific visualization. The visualization task for science communication is to show a particular environment, immerse the user in it, and communicate its characteristics.

In the context of scientific visualization, specifically in the visualization of medical datasets, Wohlfart and Hauser [WH07] presented a system that incorporates storytelling elements. They also indicate a discrete range of possible user interaction and its relation to the specific elements used to tell a specific story. They recognize several kinds of story consumption: *passive*, *playback with interactive approval*, *semi-interactive playback*, *total separation from the story*.

In the case of using visualization for science communication, the range of tasks is narrowed down. The primary purpose of such visualization is to enable comprehension of the depicted model or the whole environment. The 3D models show specific spatial characteristics, such as individual objects' shape, their distribution in the 3D space, and their abundance. The visual encoding, i. e., the mapping of the data attributes onto visual elements, is mostly fixed, pre-determined, and users do not change these mappings.

The concept of navigational modes is illustrated in Figure 3.5. The basic premise is that each of the degrees of freedom (vertical) is either *user-controlled* or *system-controlled*. Placing a DoF into the hands of the system essentially "stabilizes" it, and the user no longer has to adjust it in the navigation process. We define the individual navigational modes (horizontal) as a combination of navigational degrees of freedom placed either on the user or the visualization system side. When we directly expose a navigational degree of freedom to the end-user, they carry the responsibility of controlling these parameters. Alternatively, the navigational DoFs are controlled by the visualization system. There are many ways in which an automated approach of the system can control a degree of freedom.

The *augmentive* mode represents the first stage of aiding users in navigation. The only degree of freedom taken away from the user is the annotation. In practice, this means that the user does not have to reach for external sources to inquire about the objects visible in the visualization. They also do not need to specifically request information about these objects, by, for example, pointing with the mouse cursor to request the name of the hovered structure. The annotation is included automatically and dynamically as the user navigates the model.

The *declarative* mode stabilizes additional two degrees of freedom of the navigation. These are camera control and visibility management. The reason these two DoFs are coupled

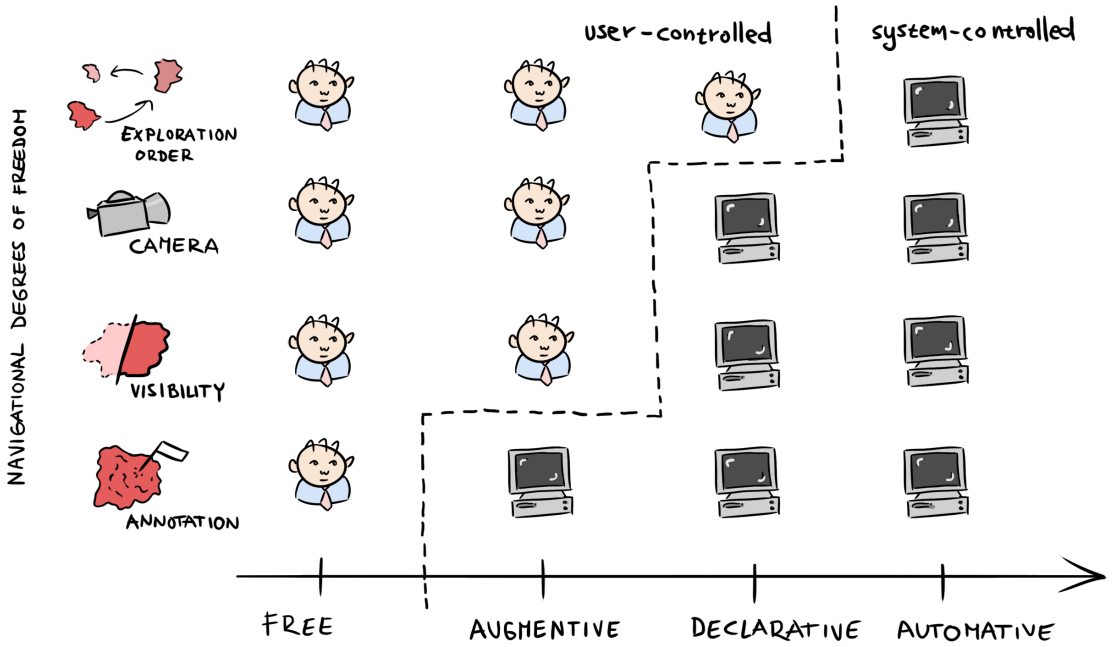


Figure 3.5: Overview of the navigational modes framework. Each of the modes—augmentive, declarative, and automative—are defined as a combination of navigational degrees of freedom given either directly into the user's hands or entrusted to an automated approach on the visualization system's side.

is the inherent occlusion present in most complex 3D models. In this environment, it does not make sense to only control the camera without appropriately adjusting the scene elements' visibility, too. The camera control degree of freedom is stabilized by utilizing procedural camera movements. The declared target determines the camera's optimal position and orientation, which is then animated to reach the computed optimal configuration. To automatically adjust the scene visibility simultaneously with the camera movement, the hierarchical information of the 3D model is used. That way, it is possible to remove the parent layers to reveal the specified target.

Finally, in the *automative* mode, the order of exploration is stabilized as a degree of freedom, too. We imagine two principal ways of stabilizing this degree of freedom and implementing the automative mode. First, the responsibility of determining in which order the elements of the model will be presented can be given to a stochastic algorithm. In such a case, the model is explored in more-or-less random order. The second option is to determine the order of exploration based on a specific storyline. Since the automative mode of navigation strongly resembles a fully guided tour where the user does not influence the walkthrough, it is natural to align the exploration structure with telling a story about the dataset.

In the rest of this section, we take a closer look at the three navigational modes and

describe our contributions to implementing these modes for visualizing biology. Afterward, the individual contributions are presented in a detailed manner in the following chapters.

3.2.1 Augmentive Navigation

Augmentive navigation builds on top of the free navigation mode in that it includes additional visual marks that help the user in the exploration. We can think of it as a variation of an information-rich virtual environment, as described by Bowman et al. [BNC+03]. An additional layer of visual information is incorporated with the spatial data. The augmentive mode can also be likened to augmented reality [Azu97]. The spatial data visualization part is replaced by a perception of the real world through a mobile device camera. This reality is then augmented by placing additional objects over the environment captured by the camera. The user has to be the one who moves through the environment—there is no way to teleport him or her like it is possible in a virtual desktop environment or even virtual reality. Not all augmented reality applications can, however, be considered augmentive navigational modes. Only those that use the augmentation to support the user in navigation, e. g., by placing labels of nearby restaurants.

This mode roughly corresponds to *orienting* and *directing* categories from Ceneda et al.'s work [CGM+17] on guidance in visual analytics. There are other examples of what we consider augmentive navigation in information visualization literature. One may think of Scented Widgets by Willett et al. [WHA07]. They augment traditional user interface elements with a *scent*, giving the user a better sense of which options can be worth selecting.

For molecular models, the work of Le Muzic et al. [LMMS+16] offers an example. In their work, the user explores the model by flying the virtual camera using the traditional controls of orbiting, panning, and zooming. They can also define cut objects to set a particular cut-away view. They include a panel that gives detailed information about the status of visibility control: They indicate the percentage of objects that were cut away for each molecular type. The inclusion of this information enables the user—a scientific illustrator—to have better control over the visibility settings.

While this enables a power user, i. e., a scientific illustrator, to be more expressive in their work, an inexperienced user could still struggle to make sense of this interface for casual exploration. The simplest way of making the environment more comprehensible for even a lay-person user is to use textual labels that annotate structures depicted in the current view. Augmentation of molecular visualization with textual labels is the topic of our first contribution to the thesis.

There are, in principle, two issues with applying conventional labeling techniques on molecular models. First, previously published methods, for the most part, count with relatively simple models with enough empty space around the 3D model. In a human body model, there will never be the case of a thousand instances of a heart, densely packed in the space. That is the case for molecular scenes, where a particular protein type can be represented in the model thousand or more times. Second, they do not label

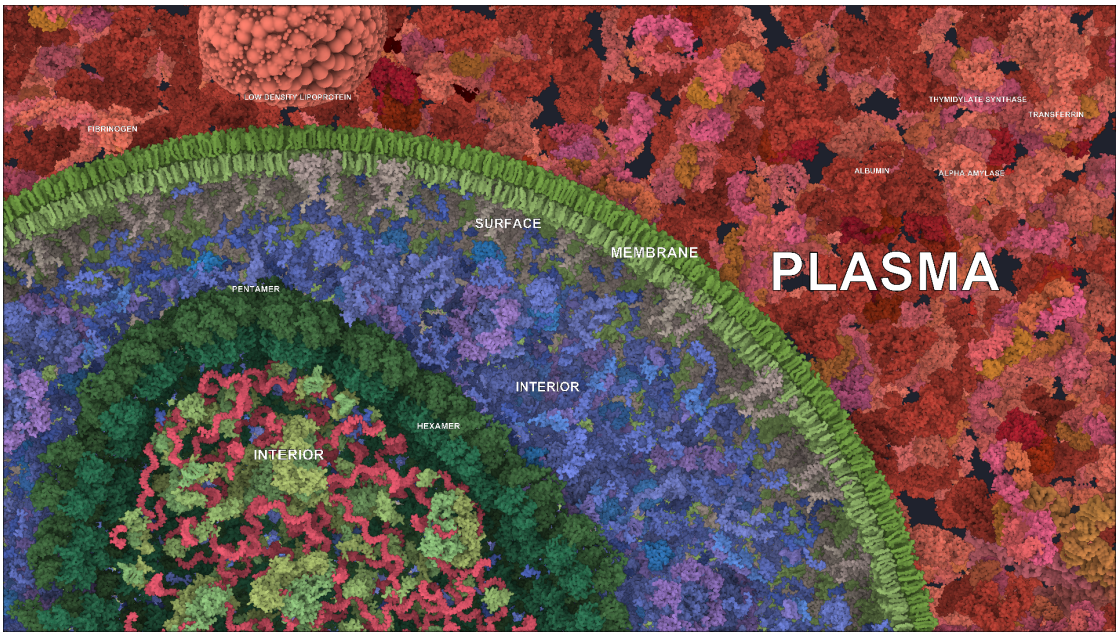
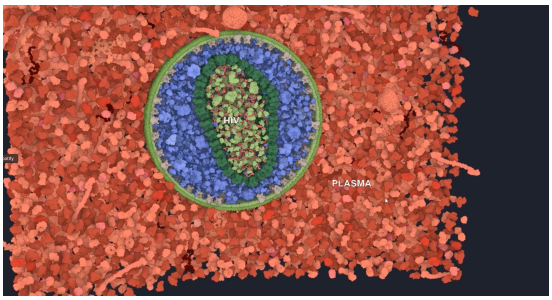
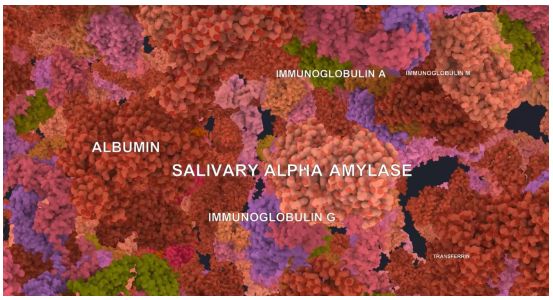


Figure 3.6: Molecular scene augmented by textual labels. Annotation on several labels can be seen in this screen capture: large-font *plasma* label denoting the whole compartment of blood plasma proteins, *membrane* or *interior* labels annotating parts of the HIV, and finally smallest-font labels marking the individual protein types.



(a)



(b)

Figure 3.7: Before and after user interaction: (a) initial zoomed-out view showing the two main components of the model (i. e., the HIV particle and its surrounding blood plasma), and (b) after the user zoomed-in onto the level of individual proteins.

multi-scale phenomena. When a number of objects can be grouped and described by a common name, a step of deciding on which level such structure should be labeled must be incorporated in the algorithm.

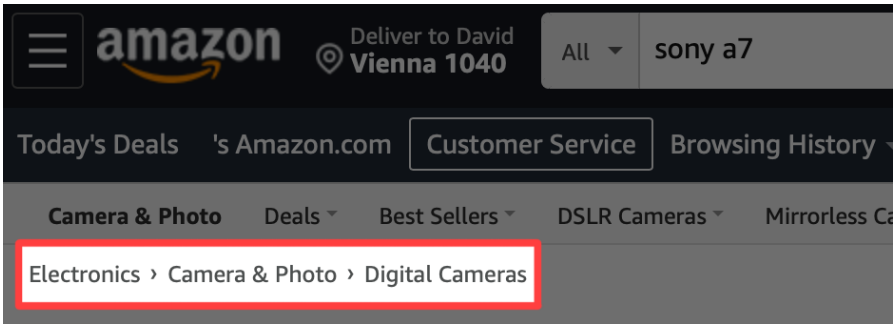
We delve into solving the technical challenge of annotation in multi-scale molecular environments in Chapter 4. We take inspiration from Level-of-Detail methods, in which a 3D model is rendered with varying levels of fidelity based on the distance from the observer. Following this idea, we determine the level of labeling, i. e., on which level of the multi-scale hierarchical organization should each object be annotated. We present a GPU-accelerated algorithm where this decision is also based on the distance from the virtual camera. With this contribution, we implement augmentive navigation for the environment built up by large molecular models.

Conceptually, labels not only function as annotation elements, but they also provide the general *entry point* for reasoning and interacting with the structures in the visualization. Even in their most primitive form, they enable further querying of information. The question "What does RNA do in the HIV?" elevates the interaction with domain knowledge instead of merely asking: "What is that red thing that looks like a rope?". Labels make it possible to be specific when asking further questions and democratize the visualization to a broader audience. They provide the vocabulary necessary for further inquiry.

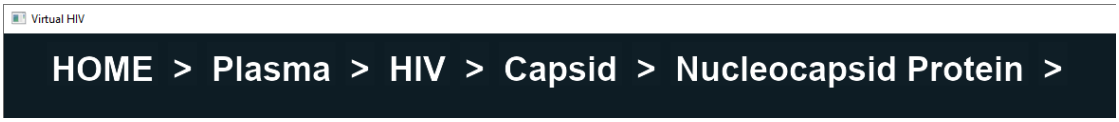
Verbal or other non-spatial information can also be embedded in a scientific spatial data visualization to orient the user in the large information space and guide them to various areas of interest. We envision that in the future, we will be able to model, render, and interactively explore virtual environments spanning whole cells, if not whole organs. In such massively multi-scale environments, where the user can too easily get lost, it will be essential to incorporate more abstract information than only textual labels annotating visible structures on the current level. In such a scenario, elements of *guidance* will have crucial importance.

As an example of orienting the user by embedding non-spatial information and, as a consequence, augmenting the navigation, we highlight the *breadcrumbs panel* from Paper B, described later in Chapter 5.

With textual labels, the information used is most often the name of a specific structure. A name is usually a good trade-off between embedding extra information and not occupying large amounts of screen space and occluding too much of the visual content. However, there is a lot more verbal information that can be supplied to the user to explain a scientific concept captured by the interactive visualization. Nowadays, a person can learn a lot about a specific topic, such as molecular and cellular biology, from various sources: online courses, textbooks, popular science articles, or even knowledge repositories such as Wikipedia. The knowledge is often tailored for a specific level of the learner. The person who wants to learn must connect many disparate pieces of information in their mind. In the traditional education pipeline, the visuals and the verbals (e. g., meanings, processes) are mostly separated, except for illustrations in textbooks. In recent years, learning through animations published at websites such as Youtube or Vimeo became more and



(a)



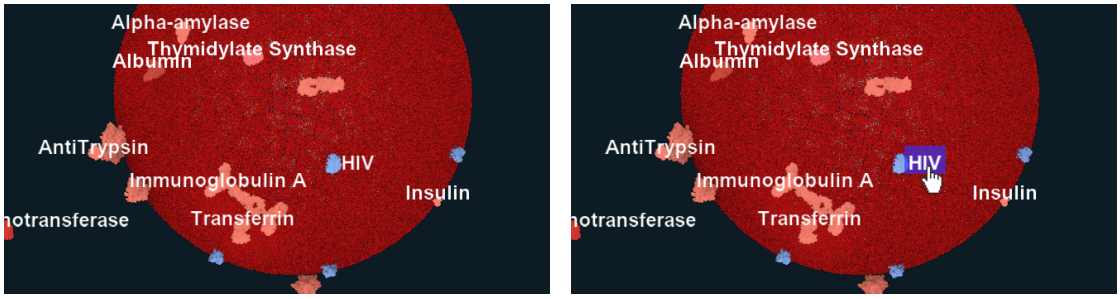
(b)

Figure 3.8: Example of an augmentive navigation element that orients the user in their exploration: Breadcrumbs panel encoding the current location in the hierarchical organization. This element is prominent in web design (a), and we adopt this design component in our navigational method (b) described in Chapter 5.

more popular. In this medium, the visual aspects are more seamlessly combined with the verbal descriptions. The issue with such videos is their laborious production. Furthermore, the creator usually takes artistic license and might not base the visuals on real measured data to make the point more understandable. Even if they present the phenomena to the best accuracy according to the current knowledge, future research findings might invalidate the depicted theories. In such a case, the whole production pipeline must be repeated from the start to the end. With real-time visualization, we have the opportunity to take the integration of current scientific knowledge in public dissemination even further. As explained by Ynnerman et al. [YLT18], interactive scientific visualization now enables to conduct science communication in the same tools and frameworks that were previously used exclusively for exploratory research tasks.

3.2.2 Declarative Navigation

Even though in augmentive navigation mode the user is given additional information that might help them to control the rest of the navigational degrees of freedom, they still carry the responsibility of finding the proper low-level parameter values to explore the 3D model. This can present an obstacle in certain venues and specific inexperienced audiences and requires additional means of making the navigation more intuitive. The second navigational mode—declarative navigation—represents this conceptual step. The idea behind this mode is not to let the user tweak all the low-level settings of each of the navigational degrees of freedom but rather have them specify an end goal or target. The visualization system then determines the low-level parameters to get the user to their



(a) (b)

Figure 3.9: Declaring the target for navigation using HyperLabels. The textual labels represent components of the model at one level of the hierarchical organization. The labels can be clicked and such action results in navigation to the corresponding component.

declared target. We can think of several cases where such a way of navigation has been used. For example, Li et al. [LRA+07] let the user select a component from a list and then adjust the cut-away settings to reveal the selected part from the complex 3D model.

We propose our flavor of declarative navigation in Chapter 5. In this work, we introduce a technique for browsing a complex multi-scale environment using so-called *HyperLabels* (Figure 3.9). We leverage the textual labels that annotate visible structures at one level of the hierarchical organization. We then make these labels active—the user can click on them, and the click indicates the intention to travel "into" the selected structure. The method is inspired by the way through which other hierarchically organized information spaces are navigated. We primarily pinpoint two examples: webpages and file systems. A file system is a more obvious connection. A file system is a hierarchically organized system of files and folders, where the folders facilitate the hierarchical encapsulation. It can be visualized as a large tree structure, where each node is either a file (leaf nodes) or a folder (inner nodes). Next, we focus on applications for navigating such a structure that use a graphical interface. There are ways to explore file systems using a command line, which are not relevant in this case. Several categories of file explorer applications exist, but most share the element where a particular folder's content is shown. Another common design element is the ability to click (or double click) on a folder in such a view, resulting in transporting the user into that folder location. This allows exploration in a top-down fashion. To travel in the opposite direction, two mechanisms are often employed: a back button or a navigatable path. Other tools can also jump between different locations: e. g., explicit specification of a new target location path or a list of favorite locations. Nevertheless, the two elements crucial to facilitate hierarchical navigation are clicking on an element to traverse onto that level and some mechanism to go back.

Tools for navigating a 3D environment are diametrically different. There the focus is, naturally, placed on the navigation of the 3D space. However, in many environments, the 3D space is also hierarchically organized, an example being the biological models.

For that reason, we couple the approach used in navigating a general hierarchically-organized information space with navigational methods for 3D spatial models. The hierarchical navigation part is facilitated by interacting with the HyperLabels, either deployed directly in the scene or placed in the breadcrumbs panel. The spatial navigation part is accomplished using the conventional orbiting, zooming, and panning interaction. These three low-level camera control mechanisms are anchored to a local coordinate system of the particular component of the hierarchical organization.

The switch to a view focusing on a particular sub-part of the model happens in a declarative manner through HyperLabels. Three visualization components are then automatically adjusted to communicate the switch of context. Camera is smoothly animated while the visibility of the model is changed to remove occlusion of the selected structure of interest. Finally, the annotation is changed by deploying new HyperLabels.

3.2.3 Automative Navigation

In the previous subsection, we described a navigational mode where a single user specifies their target by simply clicking on a textual label to indicate where they want to be taken. However, even such a simple and intuitive method can rely too much on the user in certain scenarios. For example, let us consider the case of an installation in a museum or a science center. In this case, two issues might occur. First, in such settings, it is usual that a group of people, instead of a single user, is interested in the visualization. One of them is therefore required to step up to the role of a controller for the others. If nobody assumes this role, the visualization stays in one state. Second, it is expected that this person will not be a domain expert: it could be a beginning student at a university or even elementary school-aged children. That fact makes it likely that they will skip at least some interesting or significant features in the model. The controller can also very easily get lost in an unengaging part while navigating the complex environment.

In such a case, it can be beneficial not to rely on user input for the exploration but instead leverage other information already available in guiding the exploration of the model. Through meta-data inherent to the spatial model, it can, for example, be determined which of its parts are more salient than others. Since this information is already available, it can be used in the visualization system to automatically direct the navigation without user intervention.

The idea of guided or even fully automated navigation has naturally been proposed before. Galyean [Gal95] addresses the guided navigation topic with a river analogy. The user is propelled through an authored storyline in a virtual environment similarly to how he or she would be taken on a boat by river currents. The user has means of influencing the movement via interaction—in the river metaphor user interaction maps to the control of a boat's rudder. This control has a local and a global level. It has a short-term effect in steering between the river banks, and a long-term effect when it influences which path the boat will take at a fork. Hong et al. [HMK+97] employ a similar approach in their method for interactively exploring models resulting from a virtual colonoscopy. They

simulate the virtual viewpoint as a submarine that flows inside a human colon. The user—a physician—can control the metaphorical submarine to inspect the inner surface of a reconstructured colon in search for polyps.

Algorithms for automatic navigation through 3D environments have been explored in the past, mostly in the context of determining paths of the virtual camera, and therefore disregarding the other aspect, e. g., occlusion management, discussed in the thesis. Christie and Olivier [CO09] review techniques for manipulating cameras with a focus on semi- or fully-automatic systems. They attempt to leverage cinematography concepts and translate them into the realm of real-time computer graphics.

We discuss automative navigation in the context of molecular models in Chapter 6. The basic premise is to use real-time visualization systems for the creation of content meant for presentation. As expressed by Ynnerman et al. [YLT18] through the *exploranation* term, interactive visualization has matured to the point where the visuals generated from real measured data are up to par with those authored specifically for science communication. The advantage is, however, that with real-time graphics, we can eliminate the costly production pipeline. Developing science communication materials based on real-time visualization adds flexibility in the process as it, besides other benefits, allows rapid incorporation of new knowledge.

We propose the concept of *scalable documentaries*, where real-time visualization plays a key role in providing the visual part for a scientific documentary. Direct user interaction is replaced by automated control of all the navigational degrees of freedom. Since we are located close to the right side of the interaction spectrum (Figure 3.3), the resulting visualization closely resembles a produced movie, a documentary. In a traditional documentary, there would be a script with a storyline. A storyline is one way of stabilizing the 'order of exploration' degree of freedom.

In Chapter 6, we describe a pipeline for producing *Molecumentaries*, i. e., molecular documentaries. It implements the automative navigation concept, where a molecumentary presents a molecular model without any need for user interaction. This visual aspect is coupled with a verbal commentary. The audio part is generated to eliminate the usual process of a voice actor pre-recording the commentary. We leverage modern text-to-speech software and create the voice-over commentary on the fly from textual descriptions captured from the Web. This approach reflects the flexibility in generating the visuals with real-time visualization.

The three chapters that follow describe applications and technical realizations of the three navigational modes. In Chapter 4, the augmentive navigation concept is implemented through the inclusion of textual labels in a 3D visualization of multi-scale molecular models. Chapter 5 focuses on realizing declarative navigation in the same complex environment. The textual labels used for navigation allow browsing of the 3D model by its hierarchical organization. Finally, in Chapter 6, we propose automative navigation by generating molecular documentaries that take the user on a virtual guided tour through the 3D model without an explicit need of user input.

CHAPTER 4

Textual Labeling in Complex Biological Environments

This chapter is based on the publication:

David Kouřil, Ladislav Čmolík, Barbora Kozlíková, Hsiang-Yun Wu, Graham Johnson, David S. Goodsell, Arthur Olson, Eduard Gröller and Ivan Viola, "Labels on Levels: Labeling of Multi-Scale Multi-Instance and Crowded 3D Biological Environments", in *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 977-986, Jan. 2019, doi: 10.1109/TVCG.2018.2864491.

Using modern scientific visualization methods, we can generate engaging visualizations. The visual output showing the raw spatial data is, however, not directly consumable by people who do not posses deep knowledge about the depicted domain. Domain experts must provide explanations for lay-persons and annotate the resulting visualization. Textual labeling (Figure 4.1) represents an initial step of incorporating such knowledge automatically, making the visualization able to stand alone without the need of explaining.

4.1 Introduction

New technologies and advancements in computing enable us to capture and model enormous amounts of data, describing complex multi-scale structures and processes, and store them as 3D scenes. One example of complex multi-scale structures are molecular biology models. These models represent complex assemblies of proteins where the proteins themselves are composed of polypeptide chains that in turn are composed of atoms. In this case, the scale levels are the atomic, polypeptide chain, protein, and protein assemblies levels.

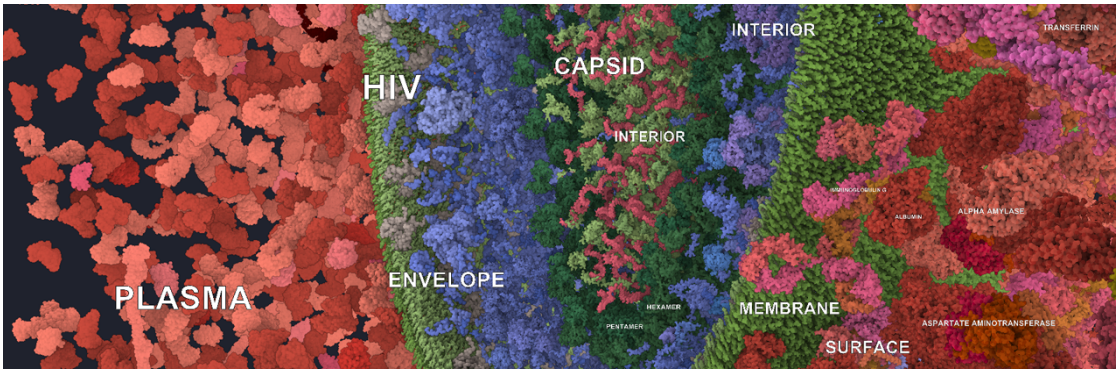


Figure 4.1: Multi-scale and multi-instance labeling of an HIV dataset that has thousands of copies (instances) of approximately 60 unique geometries: the foreground objects are automatically selected representative instances annotated with protein types, while the background objects are labeled with names of whole compartments, representing higher levels of the hierarchy.

Complex multi-scale scenes often contain millions or even billions of 3D objects. Current capabilities of modern GPUs and level-of-detail techniques enable users to explore such scenes interactively, and smoothly transition between several scale levels [Lue03]. Each of these levels shows the objects in the scene with a particular complexity, based on the distance of the object from the observer.

The multi-scale aspect of the scenes makes visual communication of their structure a challenging task. Recently, several approaches that improve the visual communication of structure of multi-scale scenes appeared. Hun et al. [HMC11] visually communicate the parent-child relationship in a multi-scale hierarchy by deforming the multi-scale scene. Waldin et al. [WLW+16] utilize a dynamic approach to color objects in multi-scale scenes based on several levels of the hierarchy to better distinguish between structures on various scale levels.

On the one hand, through visualization we are able to visually communicate spatial arrangements of the objects and, to a certain extent, their multi-scale hierarchical relationships. On the other hand, visualization cannot convey the details that are typically covered by a text description. Therefore, to ease the understanding of multi-scale structures, we need to interconnect the visualization of the complex structure with a textual description.

In this chapter, we present an approach that uses labels, i.e., textual annotations of the objects, to mutually interconnect the visualization and the textual description. The labels are positioned over the visualization of the multi-scale scene in a manner that associates them with the depicted objects. For these scenes, the labels should reflect their multi-scale aspect. In other words, we should label the objects in a scene taking scale and hierarchical grouping levels into account.

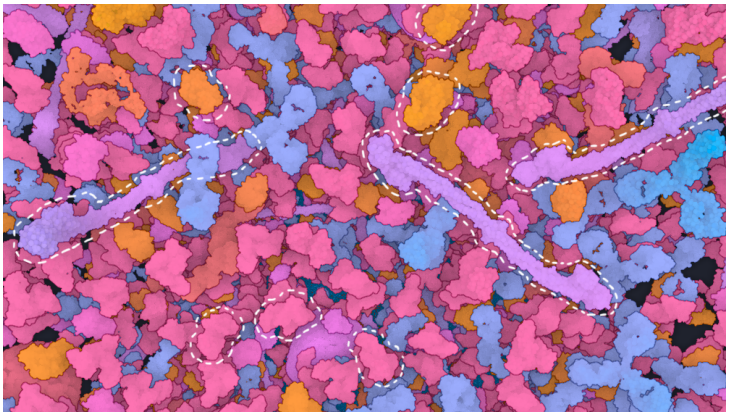


Figure 4.2: Blood plasma as an example of a biological environment. Very many instances of just a few protein types make up a dense scene. To illustrate this, three protein types and three instances each (of many) are highlighted.

Multi-scale labeling introduces several new problems that have not been addressed by existing labeling methods. For each view of the scene, we need to select appropriate scale levels on which we will label the objects. One view can contain objects from different scale levels and the labeling technique should take this into account. Changes in the scale level of the objects should induce corresponding changes of their labels as well.

Complex multi-scale scenes usually contain many objects and not all of them are unique. In our exemplary biological environment, a blood plasma can be a typical representative of such a scenario. Blood plasma is a component of blood, consisting of proteins and other molecules and ions in very dense concentrations. However, the number of distinct molecule types is very limited, and blood plasma contains many instances of the same molecules (as can be seen in Figure 4.2). Typically, there is no need to label all instances of each type, but only one or several representative instances, e.g., representative proteins of each protein type. Such multi-instance labeling requires an automated selection of the representative instances, which is not addressed by existing labeling methods.

Complex multi-scale and multi-instance scenes are typically very crowded, i.e., densely populated with objects. This holds especially for molecular biology models. Therefore, there is very little or no space where the labels can be placed without overlapping the objects in the scene. This is the case especially if the camera is close to the objects in the scene.

In this chapter, we introduce a novel solution to the problem of **multi-scale** and **multi-instance** labeling, which is applicable to arbitrary domains. We present a conceptual framework for interactive labeling in densely populated multi-scale and multi-instance environments, where annotation on several scale levels is possible and desired. The framework allows users to interactively explore complex multi-scale and multi-instance environments. It labels representative instances of objects in the environment with respect to the current viewpoint and the multi-scale hierarchy of the scene. Most importantly,

the presented method works in real-time, which allows users to interactively explore the environment. The main contributions presented in this chapter are:

- A conceptual framework of labeling in environments with novel characteristics not yet considered in previous work.
- A real-time algorithm for choosing the appropriate label level for each object in the scene. The approach allows us to label the same object type on several different scale levels.
- A real-time algorithm for selecting representative instances of a given object type for labeling. The representative instances are chosen according to visibility and position with respect to the other objects of the same type in the scene.
- A description of a label transition that happens when the scale level changes. The label transitions help the users to maintain their orientation in the multi-scale hierarchy.
- An approach to make the movement of labels temporally coherent during user interaction. This is accomplished by re-projecting world space positions of the labels and at the same time biasing the algorithm towards previous label position.

We demonstrate our solution on a scene of a human immunodeficiency virus (HIV) in blood serum [JAAA+15, JGA+14] from the domain of molecular biology. The scene is multi-scale, where the highest level contains the entire HIV virion, consisting of a lipid bilayer on its boundary and an inner compartment, containing capsid and many proteins. The capsid envelope consists of pentameric and hexameric molecular complexes and the inner part of the capsid contains proteins and a RNA fiber. Each of these structures are composed of atoms. The building parts (i.e., lipids, proteins, complexes) occur in the scene many times, making this an ideal exemplary scene for our multi-instance labeling approach. The scene is very densely populated with the objects. Despite showcasing our approach on a scene from the molecular biology domain, we believe that our approach can be utilized in other domains with similar properties as well.

4.2 Labels on Levels – Method

In this section, we describe our method of labeling multi-scale and multi-instance scenes. The method is designed to work as a post-processing step, performed after the scene has been rendered, to make it easily adaptable for use with various rendering frameworks and application scenarios. The overview of our approach is depicted in Figure 4.3.

To render the scene, we use the approach of Le Muzic et al. [LMAPV15]. Their technique allows us to render the complex multi-scale and multi-instance scene in real-time by employing a level-of-detail scheme that simplifies the shapes of proteins if they are far

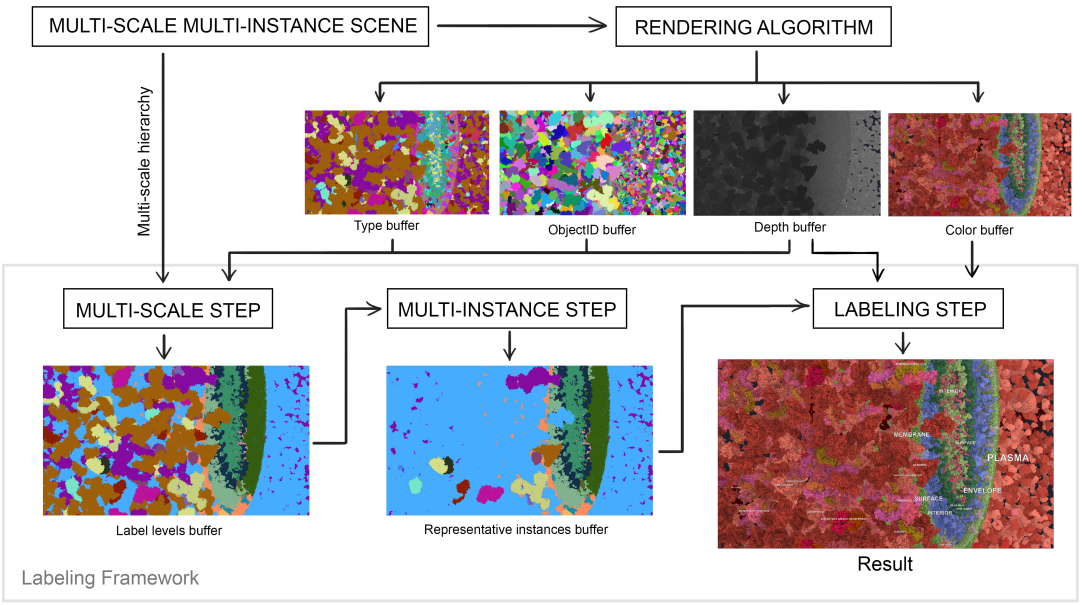


Figure 4.3: Overview of the Labels on Levels (LoL) framework. The multi-scale step utilizes the G-buffer result of a rendering algorithm with the scene's hierarchy in order to generate depth-aware regions that are to be labeled. Afterwards the multi-instance step selects a representative instance together with a label anchor point inside the chosen instance, which is optimal according to specified criteria.

from the camera. We have modified the approach to produce a G-buffer[ST90], consisting of a `Type buffer`, an `ObjectID buffer`, a `Depth buffer`, and a `Color buffer`. All these buffers, together with the multi-scale hierarchy of the scene, are the input to the Labels on Levels (LoL) method (see Figure 4.3).

The multi-scale hierarchy is supplied as a scene graph, commonly used in various 3D applications and game engines. This tree-like structure contains nodes that represent parent-children relationships between scene objects. The leaves of the hierarchy represent objects of the lowest semantic level in the scene. In our exemplary scene containing the HIV virion, the leaves correspond to proteins. The inner nodes of the hierarchy contain objects on higher scale levels composed of objects from lower semantic levels down the tree structure. Each object in the multi-scale hierarchy has a unique `objectID` and a `type` assigned. For example, if there are several instances of a certain protein type in the hierarchy, then each of these instances will have a unique object ID, but they will be of the same type. In our method we use the type information at the nodes in order to label only one representative instance of a particular object type.

The multi-scale hierarchy also expresses the spatial characteristics and extent of this particular type of data. The higher a node in the multi-scale hierarchy of the scene is, the larger the structure annotated by the corresponding label becomes. A parent node

occupies more space in the scene than each of its child nodes, because the parent node is composed of its child nodes and this should be reflected in the labeling as well.

The four buffers of the LoL approach contain various information necessary for the labeling. The `Type buffer` contains the encoded type of each protein projected onto the screen. The `ObjectID buffer` contains unique identifiers of the leaves of the hierarchy (in our case individual proteins) projected onto the screen. The `Depth buffer` contains the distance to the camera for each protein projected on the screen. The `Color buffer` contains the color of each projected protein. All buffers are implemented as 2D textures.

Our goal is to find a set of textual labels that annotate the object instances which are the most visible in the scene. At the same time, the labels should not visually clutter other objects in the scene. In addition, the movement of the labels during user interaction with the scene should be temporally coherent, i.e., the label positions should not change abruptly between frames.

We address the visual clutter by utilizing two characteristics of the data. Instead of labeling all visible protein instances in the scene, we label only selected instances on the appropriate scale levels. Our labeling framework consists of three steps:

- In the *multi-scale step*, we group pixels based on the type of the visible instance that occupies it, the distance of the instance to the camera, and the multi-scale hierarchy. After this step, each region in the output represents an instance of a specific object type on a common scale level.
- In the *multi-instance step*, we choose one instance of each object type that corresponds to the best representative for each scale level. In this step, the anchor point for the label is also chosen.
- In the *labeling step*, we label each representative instance and draw the label into the color buffer, which contains the already rendered scene. We are using internal labels that are placed in the 3D object space. Label positioning, as described in the related work, is only a small part of our method.

In the following sections, we discuss these three steps in detail, describe our approach to make label movements temporally coherent, and present our approach to the label rendering.

4.2.1 The Multi-Scale Step

The general idea behind the multi-scale step is to group objects further from the camera and label them as a group using a representative group name, rather than labeling separate objects. The distance to the camera is used to determine the appropriate level for labeling of an object in the multi-scale hierarchy. In consequence, individual objects, such as proteins, that are far away from the camera will not be labeled individually. This

Figure 4.4: (a) Illustration of subdividing the scene into several scale levels (foreground, middle-ground, background). (b) The distance of objects to the camera determines which nodes of the multi-scale hierarchy are labeled. (c) A typical camera view in the HIV scene. (d) Result of the multi-scale step for the view in (c). Here the colors indicate different object types in the `Label levels buffer`.

approach allows us to reduce the number of labels, and, at the same time, to communicate to the user in a single image the multi-scale hierarchical relationships in the scene.

Figure 4.4a illustrates the approach. In this sketch, the scene is subdivided into three regions based on the distance of the objects to the camera. In the foreground (FG, red), individual molecules will not be grouped into higher-level structures and the representative instance of each visible molecule type will be labeled. In the middle-ground (MG, yellow), the HIV virion proteins will be grouped into higher-level structures (e.g., membrane, inner matrix, and capsid) and the representative instances of these higher-level structures will be labeled. In the background (BG, green), all blood plasma proteins will be grouped into one top-level structure and annotated by a single label as *blood plasma*. In order to achieve this labeling, we propose the following algorithm.

The input to the multi-scale step is a G-buffer consisting of a `Depth buffer`, an `ObjectID buffer`, and a `Type buffer`. These buffers are determined during the rendering of the scene. An additional input to the multi-scale step is the scene graph, which contains the multi-scale hierarchy of the scenery. The input G-buffer contains depths, object IDs, and types of the objects at the lowest scale-level, which are stored at the leaves of the multi-scale hierarchy.

The output of this step is the `Label levels buffer`, a G-buffer containing the distance of objects to the camera, instance IDs and types of the objects represented by nodes of the multi-scale hierarchy at the determined labeling levels.

In the preprocessing step, we calculate the number of nodes on the path from each leaf node to the root in the multi-scale hierarchy. The appropriate labeling level is determined for each pixel of the input G-buffer in parallel. The algorithm for one pixel works as follows:

Step 1 We obtain the distance to the camera, the unique object ID, and the type from the input G-buffer.

Step 2 Based on the object ID, we divide the $[0, 1]$ depth range into n uniform intervals, where n is the number of nodes on the path from the object node to the root of the multi-scale hierarchy (see Figure 4.4b).

Step 3 We traverse these intervals $I_0, I_1, I_2, ..., I_{n-1}$. For each interval I_i:

1. We test if the distance of the visible object at the pixel to the camera (in the $[0, 1]$ range) is in the interval I_i.
2. If the distance is not in the interval I_i, we continue with the next interval I_{i+1}. We take the interval I_i if it is the last one.
3. Otherwise, we return the i^{th} node (its instance ID and type) on the path from the leaf to the root as the appropriate labeling level for this pixel.

The output `Label levels buffer` has the same size as the input and for each pixel contains the instance ID and the type of the object (node) determined by the above algorithm. Further, we store the distance of the object to the camera in the `Label levels buffer` to speed up reading of the data in the multi-instance step. An example of the result of the algorithm for the scene in Figure 4.4c is depicted in Figure 4.4d. Here the objects in the scene are colored according to their types.

4.2.2 The Multi-Instance Step

In the multi-instance step, we are searching for the representative instance for each object type in the `Label levels buffer`. We are evaluating the pixels of all instances of each object type according to various relevance criteria described below. For labeling we are choosing the instance covering those pixels that satisfy these criteria the best. We use the position of the most suitable pixel for the anchor point. The anchor points will then be utilized in the labeling step to place the labels. The input for this step is the `Label levels buffer`, containing the instance IDs and types of objects on the determined labeling levels together with their distances to the camera. The output of this step is the `Representative instances buffer`, i.e., a 1D texture where each texel contains the information about a label placement in the image for one type of object each. In the color components of the texel, we store the instance ID and type of the determined representative instance that will be annotated. The buffer also contains the 2D screen-space coordinates of the labels.

We evaluate the pixels according to four criteria:

Saliency criterion C_1: We use internal labels to annotate the objects in the scene. To allow the users to easily associate the label with the labeled instance, we determine the instance that is the most salient, i.e., the visibly most prominent, one. In such a case, the internal label is more likely to be completely inside the projection of the labeled instance and will not occlude other objects in the scene. We model the saliency of a pixel with respect to an instance as the image-space distance from the instance silhouette. This corresponds to the largest non-occluded segment of the projected instance onto the image plane

$$C_1 = dist(\vec{p}, S), \tag{4.1}$$

where $\vec{p}$ is the position of the pixel, S is the silhouette of the instance, and the function *dist* returns the shortest distance of the pixel from the silhouette in image space. A pixel farther away from the silhouette is more salient than one closer to the silhouette. The instance with the most salient pixel is then the most salient instance.

Distance criterion C_2: Generally in labeling, one of the main requirements is that the labels should be clearly readable. As we are positioning the labels directly in the 3D space of the scene as screen-aligned billboards, the labels should not be occluded by the labeled instance itself. To achieve this, we need to assign the label to the pixel of the

instance that is the closest to the camera. We model this criterion as

$$C_2 = d_p, \tag{4.2}$$

where d_p is the pixel's distance from the camera in the range $[0, 1]$. This criterion penalizes the pixels of an instance that are further away from the camera. Again, the instance containing the pixel closest to the camera itself is also the closest one to the camera.

Border criterion C_3: If a label is close to the border of the screen then part of it may be outside, making the label unreadable. To avoid this, we want to assign the label to an instance that is not close to the border of the screen. First, we calculate the minimal distance of the pixel from the screen border as

$$d_B = min(x, 1 - x, y, 1 - y), \tag{4.3}$$

where we assume that the screen is transformed to the unit square and (x, y) is the position of the pixel on the unit square screen. We model the border criterion as

$$C_3 = \begin{cases} 1 : d_B \geq T_B \\ d_B : d_B < T_B, \end{cases} \tag{4.4}$$

where d_B is the distance of the pixel from the border of the screen and T_B is a distance threshold. All pixels closer to the border of the screen than the threshold T_B will be penalized. In our implementation, we use the experimentally determined value $T_B = 0.2$. However, this value can be adjusted by the user. The optimal value of the threshold depends on the size of each label. An automatic calculation of the optimal value for each label will be part of future work.

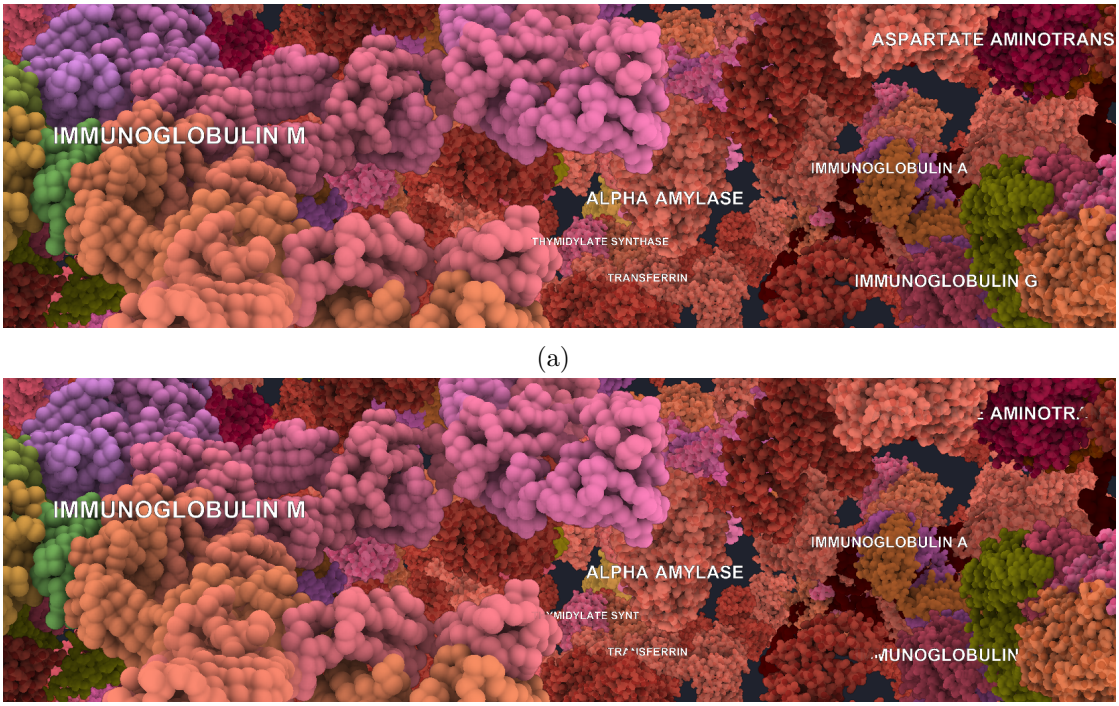
Temporal coherence criterion C_4: Abrupt changes of label positions during interaction distract the user and require him or her to mentally associate labels with the labeled objects again. Therefore, we reduce abrupt changes by positioning the labels for the current frame close to their positions in the previous frame.

Let us assume that $\vec{p}$ is the position of the pixel, $\vec{a}$ is the anchor point of the representative instance of the same object type as the object type of the pixel in the last frame, and function $dist(\vec{p}, \vec{a})$ returns the image space distance between $\vec{p}$ and $\vec{a}$. The temporal coherence criterion is then modeled as

$$C_4 = \begin{cases} 1 : \vec{a} \text{ does not exist in the previous frame} \\ 1 - min(1, dist(\vec{p}, \vec{a})) : \vec{a} \text{ exists in the previous frame.} \end{cases} \tag{4.5}$$

In consequence, pixels far away from the anchor point of the same object type in the last frame will be penalized, if such an anchor point exists.

Our algorithm for the pixel evaluation works as follows:



(a)

(b)

Figure 4.5: Comparison of the two label rendering modes. (a) The labels are rendered over all objects in the scene. (b) The depth test with the objects in the scene is performed for the rendered labels. The depth test can enhance the perception of the labels as 3D objects and also help with the association between the labels and labeled objects.

Step 1 We detect silhouettes as discontinuities of the IDs in the `Label levels buffer` and store these in a 2D texture.

Step 2 We calculate the distances from the silhouettes of Step 1 with the jump-flooding algorithm [RT06] and store them in a 2D texture. The texture contains the value of the C_1 criterion for each pixel.

Step 3 For each object type, each pixel is evaluated according to the criteria C_1, C_2, C_3, and C_4. Criterion C_1 is obtained from the texture created in Step 2. Criterion C_2 is obtained from the `Label levels buffer`. Criterion C_3 is calculated from the pixel position using Eq. (4.4). Criterion C_4 is calculated using Eq. (4.5) from the pixel position and the preceding anchor point $\vec{a}$ obtained from the `Representative instances buffer` of the previous frame. For each pixel we aggregate the individual, often contradicting, criteria using multi-criteria fuzzy decision making by Bellman and Zadeh [BZ70]. All criteria are modeled as membership functions where the values are always in the range $[0, 1]$. A value indicates the membership of the pixel in a fuzzy set,

0 means that the pixel is not in the set and 1 means that the pixel is entirely in the set. To aggregate the criteria C_i to a single scalar C (a higher value is better), we use a non-compensating aggregation with natural fuzzy conjunction. This corresponds to a simple multiplication, where the dissatisfaction of one criterion cannot be compensated by the satisfaction of other criteria:

$$C = \prod_{i=1}^{4} C_i. \tag{4.6}$$

The aggregation of the criteria for pixels where all the criteria are only partially satisfied will yield a higher C value than the aggregation of the criteria for pixels where even only one criterion is dissatisfied. This corresponds with our intentions, as placing labels on positions where at least one of the criteria is dissatisfied leads to bad label layouts.

Then, we select the pixel and the instance containing the pixel with the highest C value where all individual criteria are well satisfied. This step is done in parallel for all pixels with scattering [SH07]. The result is stored in the `Representative instances buffer`, a 1D texture where each texel contains the position of an anchor point, i.e., the position of the pixel with the highest C value, for one object type.

4.2.3 The Labeling Step

We utilize internal labels that are placed over the annotated objects. The internal labels are associated with the annotated objects through proximity. Such an association is typically weaker than in the case of external labels, which are explicitly connected with the annotated objects through leader lines.

Our decision has been influenced by two factors. First, densely populated multi-scale and multi-instance scenes do not contain enough free space where the labels could be positioned without occluding any other objects. Second, we see the weaker association of internal labels with the annotated objects as an advantage for structures on higher scale levels, which are represented by the inner nodes of the multi-scale hierarchy. The leader line of an external label points to exactly one position in the image. However, if labeling a compartment composed of dozens of instances of different object types, the leader line will point only at one of the instances. The viewer is likely to associate the label with this particular instance instead of the whole compartment. In such a case, the association can be misleading.

Label placement is done similarly to Bell et al. [BFH01] who position labels over the tagged objects. They position a label at the center of a bounding box enclosing the tagged object. Instead, we position the center of each label at the corresponding anchor point. We obtain the anchor points from the `Representative instances buffer` produced by the multi-instance step.

The output of the multi-instance step is a 2D position for a label. We now reconstruct the 3D position of the label using the camera parameters and the depth buffer [Pet09].

Each anchor point is a position in 3D and we employ *billboards*, i.e., rectangles in 3D that are always aligned to the camera, as canvases for the labels. The use of billboards positioned in 3D improves the association of the labels with the annotated objects and allows us to make the movement of the labels temporally coherent during interaction. Additionally, the 3D billboards have been received positively by our collaborating domain experts.

4.2.4 Label Rendering

In this section we present our approach to rendering of the labels. We further discuss how the various scale levels are visually communicated through the labels.

We render the labels as camera-aligned billboards. The text is placed on the billboards with texture mapping. In our implementation we are using the FreeType library [TWL] to access font character data as textures. The center of each billboard is aligned with the position of the associated anchor point that has been obtained from the multi-instance step.

We support two modes of label rendering. In the first mode, the labels are placed over the input `Color buffer` where they are occluding each object beneath. In the second mode, we perform a depth test for a label with all objects in the scene except the tagged one. The depth test is executed based on the information in the `Depth buffer`. In the second mode, a label can be occluded by the objects in the scene. However, this occlusion enhances the perception of the label as a 3D object, which in consequence helps the user to associate the label with the annotated object. Examples of both label rendering modes are presented in Figure 4.5.

Labels positioned in one frame are typically tagging objects at various scale levels of the multi-scale hierarchy. Therefore, we need to visually communicate the different scale levels of labels. We apply the concept of a visual hierarchy from graphic design. The size of the labels encodes the scale levels as this naturally implies ordering of the scale levels, i.e., a bigger label is annotating an object on a higher scale level than a smaller label. However, in our case we encounter one complication. As the labels are 3D objects positioned in 3D, the size of the labels is affected by perspective foreshortening. This results in a conflict in the interpretation of the perceived size. The size of the label indicates both the distance from the viewer and the scale level of the tagged object. After discussing with several domain experts, we address this issue by setting appropriate size ranges for each scale level. These can also be adjusted by the user.

Using size to communicate the hierarchy is the most natural option in 2D environments (e.g., text processing), but this is not directly transferable to 3D. Other options, such as different font faces, colors, or glyphs assigned to labels on different levels, are also problematic as they require an explicit mapping to scale levels. It might be beneficial to combine several approaches, e.g., assigning glyphs to different levels in addition to the font size scaling. Further, the hierarchical relationship could be communicated with lines

connecting a parent with its children. To avoid visual clutter and edge crossing such lines would only be shown on demand, e.g., during interaction with the associated label.

4.2.5 Temporally Coherent Movement of Labels

In interactive applications exploration is an important component. For labels a temporally coherent movement is crucial. Label placement frame by frame without taking temporal coherence into account can induce abrupt changes in label positions. Even a small change of the camera position or angle may cause a significant change in the input buffers with a strong impact on label positioning.

In this section we discuss approaches to smoothly adjust label positions when the user explores the scene. We employ two strategies to achieve the temporally coherent movement of the labels.

Biasing Towards Previous Results

A popular strategy to ensure that the label position will not be significantly different between successive frames is to steer the algorithm of label placement towards choosing the same (or a similar) label position as in the previous frame. However, the algorithm should not prefer positions of the labels from the last frame if this would result in a labeling that is unfavorable with regards to the other label placement criteria. In our method, the preference for the previous frame's label positions is expressed by the temporal coherence criterion C_4. As the aggregation of the individual criteria in Eq. (4.6) is non-compensating, the temporal coherence criterion cannot compensate for deficiencies concerning the other criteria. Therefore, label positions close to the label positions in the last frame will not be selected if one of the criteria C_1, C_2, or C_3 is bad for such positions in the current frame.

This strategy results in labels with a floating behavior, i.e., the labels seem as if they are attached by springs to the tagged objects, with occasional abrupt changes of their positions. The floating appearance of the labels might distract the user during heavy interaction with the scene. To further stabilize the movement of the labels, we use a second strategy during interaction, i.e., *label anchoring*.

Label Anchoring in 3D

The approaches described in Section 4.2.1 and Section 4.2.2 produce 2D label positions as output. If we would use this output in each frame, upon user interaction the labels would slightly move around in a jittering motion. This would happen even if we incorporate the temporal coherency strategy described in Section 4.2.5.

The idea behind label anchoring to further stabilize the movement of the labels is straightforward. When the camera changes, instead of calculating the labels of the current frame, the computed labels from the last frame before the interaction are “anchored" at their 3D positions on the surface of the model. As described in Section 4.2.3, we

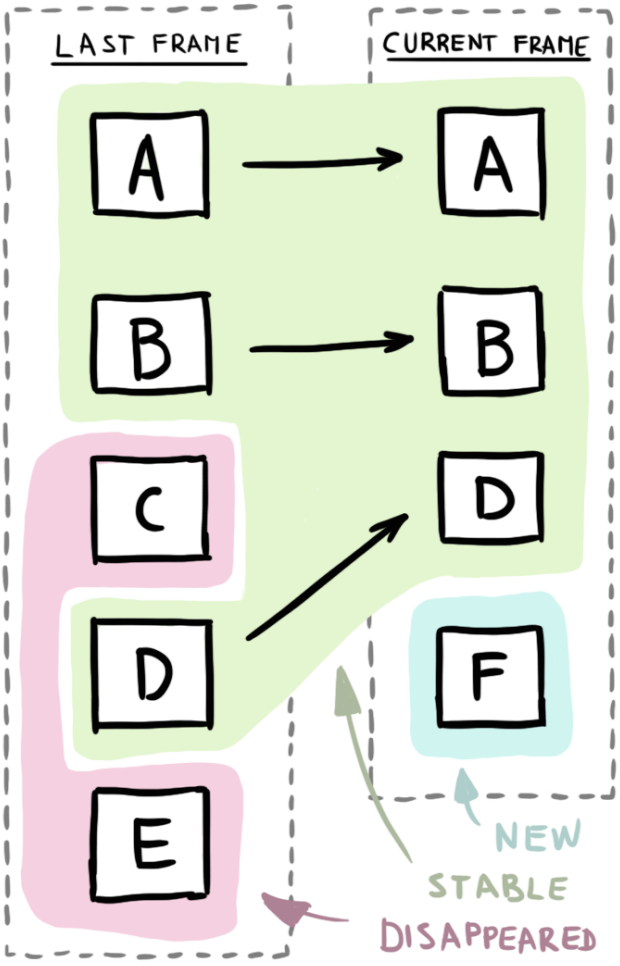


Figure 4.6: Matching of labels across two frames.

determine the 3D position of the label on the surface of the annotated model. During interaction, we re-use the anchored 3D positions and perform the label rendering as described in Section 4.2.4 with the anchored labels instead of newly computing them. A similar approach for external labels has been introduced by Tatzgern et al. [TKGS14].

Label Transitions

In this section we describe label transitions. These occur if a new label appears in the current frame, if an old label disappears in the current frame, and even if a label exists in both the previous and the current frame but on different positions. Such a situation occurs when the user stops interacting with the scene. Then, we need to make transitions from the *set of anchored labels* to the *set of the new labels* calculated for the current

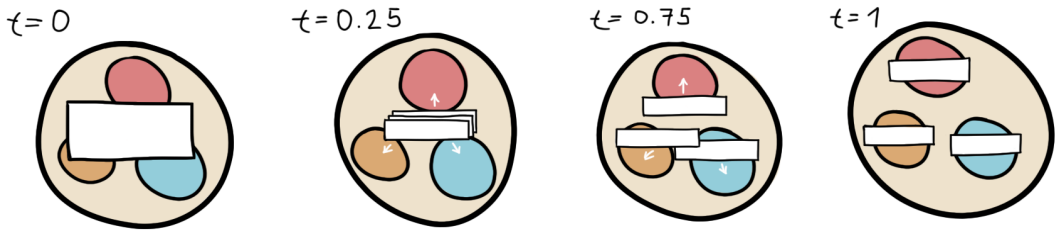


Figure 4.7: Transition from one parent label on a high level to several child labels on a lower level.

frame.

We examine the differences between the two label sets and categorize the labels in the two sets into three categories (see Figure 4.6):

New label is in the set of new labels and is not present in the set of anchored labels.

Stable label is in the set of new labels and also in the set of anchored labels.

Disappeared label is in the set of anchored labels and is not in the set of new labels.

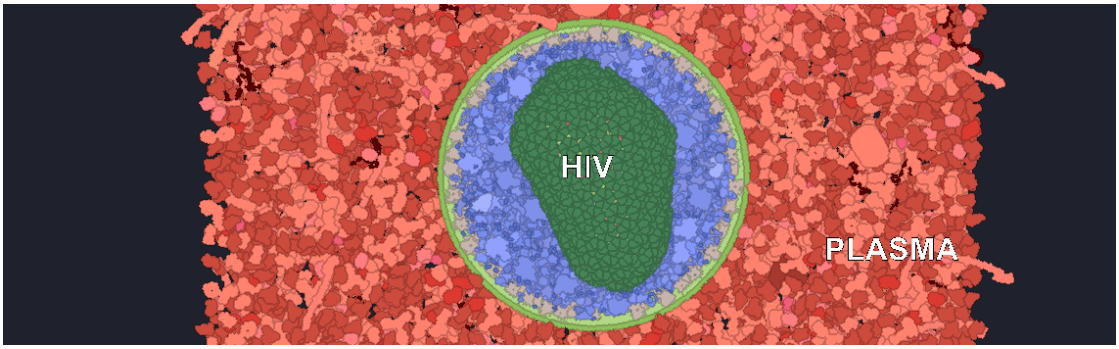
By making this distinctions, we can communicate the changes in the state of the labels. New labels are not shown right away, but instead, they are faded-in using the alpha component of the label colors. Similarly, the disappeared labels are not hidden immediately, but they are faded-out slowly. The transitions for stable labels include an animation where the labels travel to their new positions over the course of few consecutive frames. These labels however should not be far away from their anchored positions due to the biasing towards previous results. This movement can be customized using the easing function to provide a more coherent perception.

There are further possibilities for transitions between the two label sets. For example, if a parent label crosses a scale level, several child labels should be rendered instead of the parent label. An animated transition that highlights this relationship can be performed, as illustrated in Figure 4.7.

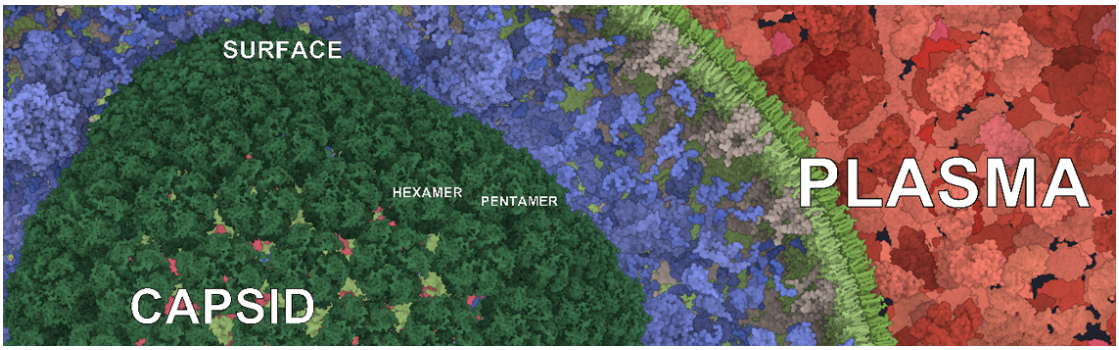
4.3 Results and Limitations

An implementation of the presented technique has been realized in the Marion framework for communicating biology [MKS+18]. We specifically extended the part of Marion dealing with large models from mesoscale biology. Real-time performance has been achieved using a modern graphics card (NVIDIA Titan X).

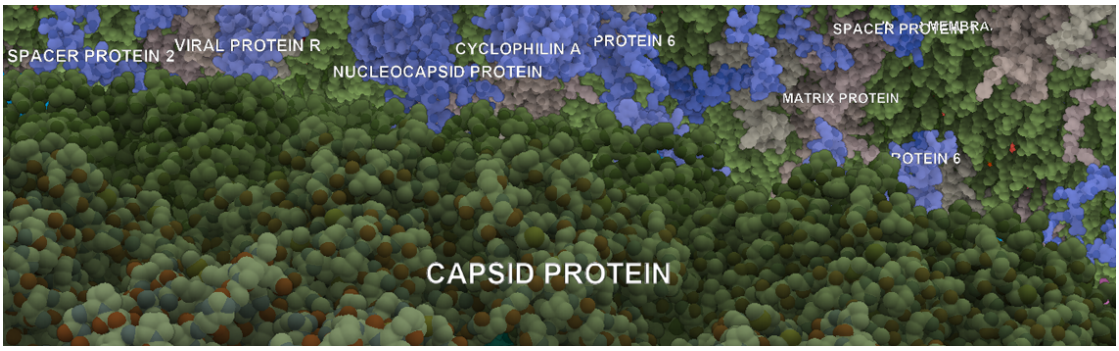
We demonstrate the benefits of labels on levels with a large scene representing an HIV virion immersed in blood plasma. The scene consists of more than 30k protein instances



(a)



(b)



(c)

Figure 4.8: Labeling the capsid of the HIV virion. The capsid is one of the inner compartments of the virion, protecting the RNA fibre. The capsid surface is composed of hexamers and pentamers, which in turn consist of capsid proteins.

from 39 distinct types (excluding the membrane lipids). The virion consists of the envelope, made up by a lipid bilayer, and an inner matrix, formed by proteins and a capsid. The capsid envelope consists of pentamer and hexamer complexes, formed by proteins, and the capsid inner part contains proteins and the RNA fibre. Especially labeling an instance of both a hexamer and a pentamer is important, as these complexes look alike and might be considered the same on a first sight.

The multi-scale hierarchy of this scene contains six scale levels, where we exclude the root from labeling. The root of the hierarchy consists of two parts, i.e., the blood plasma and the HIV virion. Each of these contains a different number of hierarchical layers. Whereas the blood plasma consists only of one additional scale level containing all its proteins, the HIV virion consists of five different scale levels. Figure 4.8 illustrates a part of this hierarchy with the corresponding labeling. It also clearly shows that the hierarchy level corresponds to the label size. The interaction with the scene and transitions between labels are demonstrated in the supplementary video[1].

Strict thresholding for categorizing pixels into levels based on depth produces labels that pop in and out frequently if objects cross the threshold during interaction. This could be solved by processing the multi-scale regions by a mathematical morphology operator. With morphological closing, certain small regions that leak into the depths on another level will be removed and the threshold value is that way used in a more fuzzy way. In order to improve the temporal coherence hysteresis threshold could be applied as well.

Further, the labels occasionally overlap and thus readability is reduced. This is due to the approximation of labels as point data in our algorithm. Such an approximation is insufficient especially for labels containing longer text. To eliminate the overlaps, we need to handle the labels as rectangles instead of points. Another limitation of the current solution is that a label occludes the tagged object itself. This can be improved by enhancing the interaction. If the user hovers over the label, it is automatically repositioned to reveal the underlying tagged object of interest.

4.4 Discussion

We showed Labels on Levels (LoL) applied on mesoscopic biological data to our collaboration partners from structural biology. In general, their feedback has been positive, indicating that in structural molecular biology there are few labeling methods available. Little formal work has been done to improve on those. Nowadays, there are many efforts to generate multi-scale data and to assemble decades worth of results into whole cell models. Our collaborating biologists believe that labeling will be important in annotating these models.

A lot of attention in our discussions with domain experts has been dedicated to the communication of scale through label size. It seems to be a hit-or-miss situation, whether people understand this relationship or if they are confused by the double mapping (hierarchy level, perspective foreshortening) described in Section 4.2.4. This is an issue we will definitely need to address in future work.

There are now projects that build models ranging from whole organs, such as the human brain (with decimeters in size), down nine orders of magnitude to the atoms in the molecules making up the brain cells. We need to adapt the current visualization and labeling techniques to these massive multi-scale characteristics. Already a mesoscale

[1] Available on Vimeo: `https://vimeo.com/513393056`

model is posing visual problems that are difficult due to the model's hierarchical structure, size, and complexity. The rather simple labeling methods available in most molecular viewers are of little use. Labels on Levels is the first attempt to simultaneously annotate objects at various scale levels. This presents the opportunity that the method will be an essential tool in scale-dependent exploration, providing identification cues that work in concert with more general perceptual hints, such as color and molecular shape.

In the context of the whole thesis, the Labels on Levels method represents the first step towards aiding with navigation of the complex biological environment. It is also the first layer of augmenting the purely spatial data with additional information. This new layer of visual information presents an opportunity for basing the interaction with the 3D scene in our later work. Without the labels it can be ambiguous to determine, using just a mouse pointer, what structures the user intends to select. The actual elementary components rendered in the scene (i. e., molecules) can build up a larger component, in which case a selection of one particular molecule can be interpreted as either selection of the low-level or high-level object.

In the next chapter, we present a method for intuitive navigation of the 3D model where the textual labels provide an interface between the user and the data. The labels act as interactive elements used to navigate the 3D space and across the label levels.

CHAPTER 5

Declarative Navigation of Hierarchical Spatial Models

This chapter is based on the publication:

David Kouřil, Tobias Isenberg, Barbora Kozlíková, Miriah Meyer, Eduard Gröller and Ivan Viola, "HyperLabels: Browsing of Dense and Hierarchical Molecular 3D Models", in *IEEE Transactions on Visualization and Computer Graphics*, 2020, Early Access, doi: 10.1109/TVCG.2020.2975583.

In the previous chapter, we showed how the annotation can be dynamically adjusted while the user navigates the 3D model. The other navigational degrees of freedom had to be controlled by the user. They had to, for example, fly the virtual camera. The cut-away has also been set up in a way that essentially showed a cross section to include all the model's levels. This cut has been determined manually and it led to a loss of information by cutting away all the objects in front of the cut. To truly explore the hierarchical and dense molecular model, the user would have to simultaneously adjust the visibility while flying the camera. The hierarchical organization of the model is also not properly communicated: the user would, for example, have a hard time figuring out of how many and what elementary components is the depicted virus composed.

In this chapter, we present a method for navigating the 3D model in an intuitive way, suitable for a wide range of audience, even those with very little domain knowledge. The camera control and visibility adjustment is determined procedurally based on the user's specification of their intended target. Textual labels, introduced in the previous chapter, act as the interface for selection of the user's target.

5.1 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 [MKA+19, MKK+19]. Two such complex systems have been modeled at The Scripps Research Institute using cellPACK procedural modeling [JAAA+15], i. e., a model of HIV [JGA+14] (Figure 5.1) and a draft model of the Mycoplasma bacterium (Figure 5.9). 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;

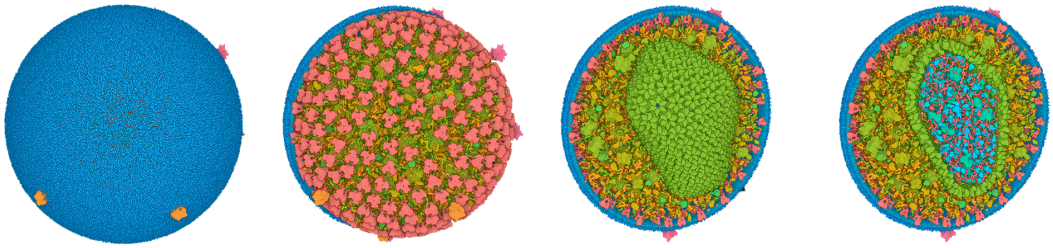


Figure 5.1: The organization of models from structural biology can be complex and it is often impossible to convey its architecture using just a single view and cutting settings. Here, we consecutively open up a model of an HIV particle to show its several levels, each communicating a different spatial subpart of the virus.

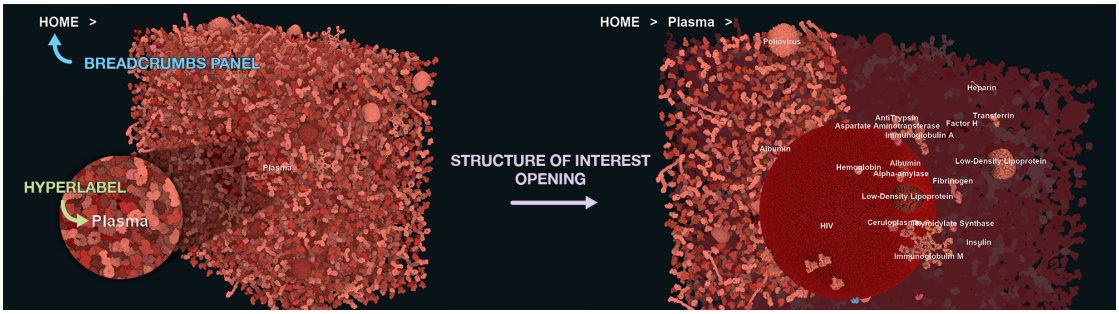


Figure 5.2: Overview of the three components of our method: HyperLabels trigger the operation of Structure of Interest Opening, which adjusts the visibility of the model to allow further exploration. This way the model is traversed in a top-down fashion. The breadcrumbs panel provides orientation and enables navigation in the opposite, bottom-up, direction.

- 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.

5.2 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. [MMGK09], 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.

5.2.1 Abstract and Spatial View Coupling

The meta data containing the information about hierarchical relationships of individual parts of the model represents a 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., [BVG10]) 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 [Kru05, 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.

5.2.2 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.

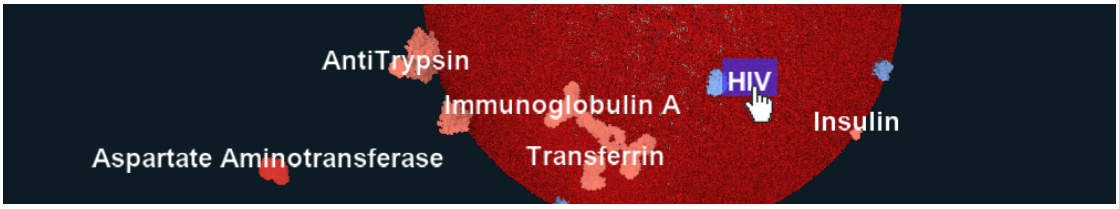


Figure 5.3: HyperLabels are textual visualization elements that combine annotation with interaction. Here, HyperLabels are placed in the scene to enable navigation in the spatial model. A blue background is displayed for a HyperLabel hovered over, to indicate, together with the changed cursor, that the text is clickable.

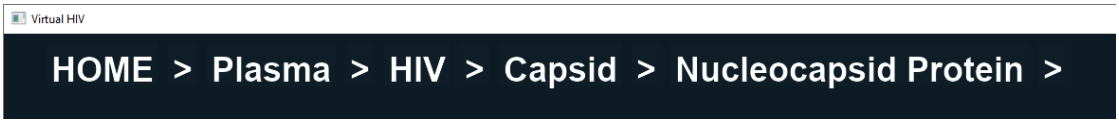


Figure 5.4: Inspired by a similar concept used in web design, the breadcrumbs panel provides orientation in the hierarchical organization of a 3D model. Interacting with the breadcrumbs panel facilitates the traversal back, towards higher-level structures in the model. We position the panel in the top left corner of the 3D spatial view, thus integrating a minimal expression of the hierarchical information to the user.

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 5.3).

The primary interaction purpose of HyperLabels is, therefore, to establish the upcoming focus: A click on a HyperLabel 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.

5.2.3 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 5.4), 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 [Kru05]. 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.

5.2.4 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 5.3.1.

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 5.3.2.

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 5.3.3.

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 [HR07]. 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 HyperLabels. 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.

5.3 Technical Realization

The exploration usually starts at the highest level of the hierarchical organization. We thus render the whole 3D model [LMAPV15] 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.

5.3.1 Sparsification

In contrast to other smart visibility techniques [VG05] 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.

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.

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* sparsification (Figure 5.5, 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 5.6 where the removed parts are indicated. To counter this weakness, we offer a second strategy—*context-cut* sparsification (Figure 5.5, 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 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.

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., [TIP05]) 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 5.5.

To demonstrate this concept (see Figure 5.6), we implemented a *ghosting* effect [FS92] which does not completely hide the cut-away objects, but rather renders them with a lower opacity. We also show a *contour* style (e. g., [BH19, IFH$^+$03]) where only a line representation of the structure of interest is drawn.

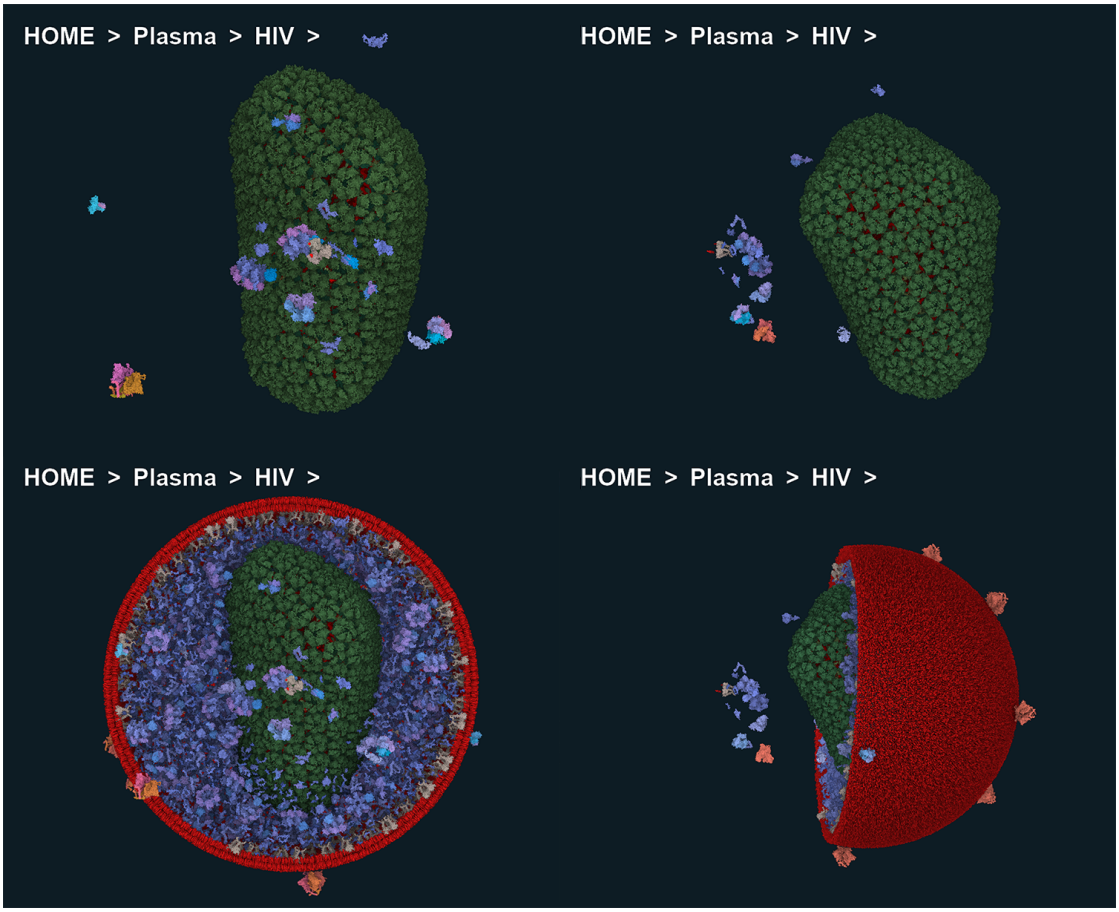


Figure 5.5: Comparison of the two implemented sparsification strategies: approach where only exactly one instance for each object on the current level is kept and shown (top); the *context-cut* method where parts of the model behind the farthest representative instance are still shown, by which we provide more contextual information about the spatial organization of the model (bottom).

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.

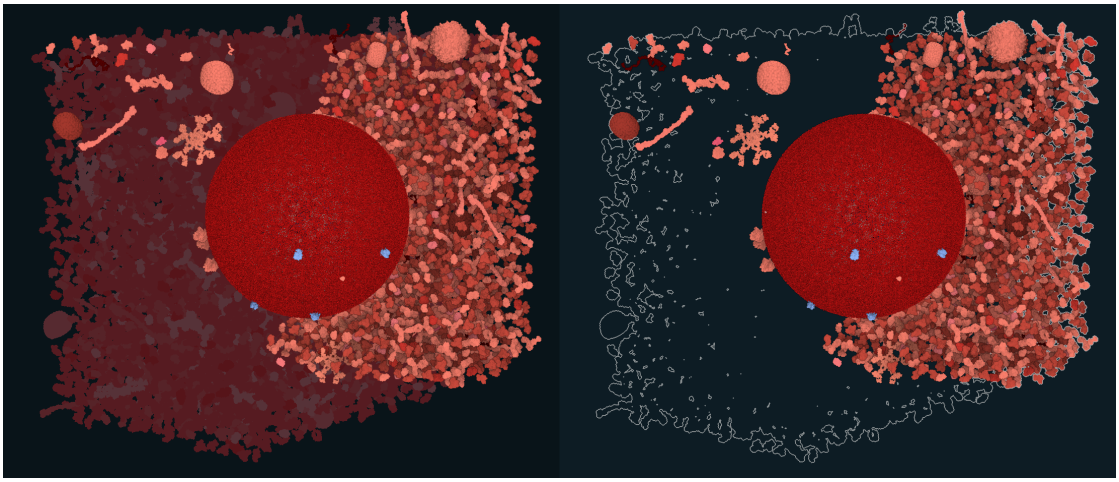


Figure 5.6: Parts affected by the sparsification can be rendered with various styles: they can be completely removed from the scene, displayed using a ghosting effect, i. e., rendered with a lower opacity (left), or rendered as only contours (right). Contouring and ghosting indicate the cut-away parts and thus provide a context information to the user.

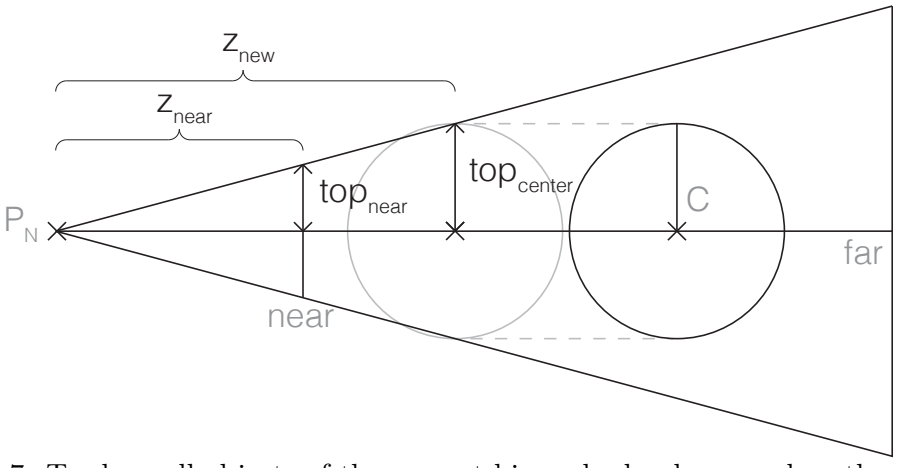


Figure 5.7: To show all objects of the current hierarchy level, we enclose them with a bounding sphere. We then move the camera such that the view frustum touches the top (and bottom) points of the bounding sphere.

5.3.2 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:

$$P_N = C + z_{new} \cdot |P_C - C|, \tag{5.1}$$

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 5.7) as:

$$z_{new} = \frac{z_{near} \cdot top_{center}}{top_{near}}. \tag{5.2}$$

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.

5.3.3 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 [KČmolíkK+19] 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. [KČmolíkK+19] 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.

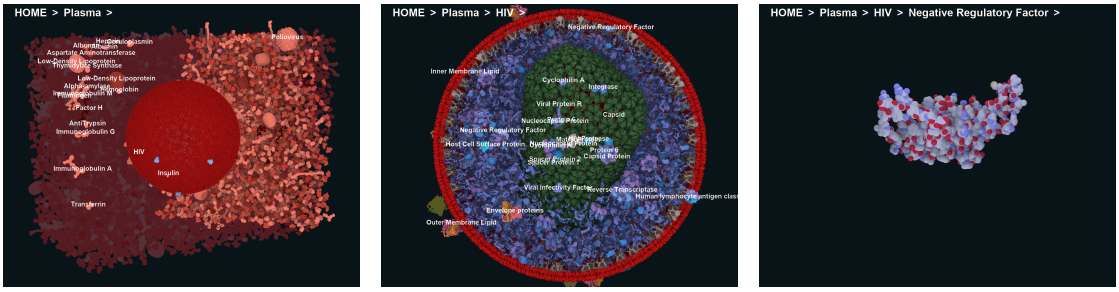
We introduce those HyperLabels that are newly added to the scene by fading them in, while those that are removed from the scene are faded out. Later, it is possible for users to explore the focus structure locally using orbiting, panning, or zooming. To facilitate such exploration, we fade out HyperLabels 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.

5.4 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 [JAAA+15] and our proof-of-concept implementation is built on top of the Marion framework [MKS+18]. We show the HyperLabels interaction in the supplemental video[1], where the textual labels can also be observed in a sufficient size.

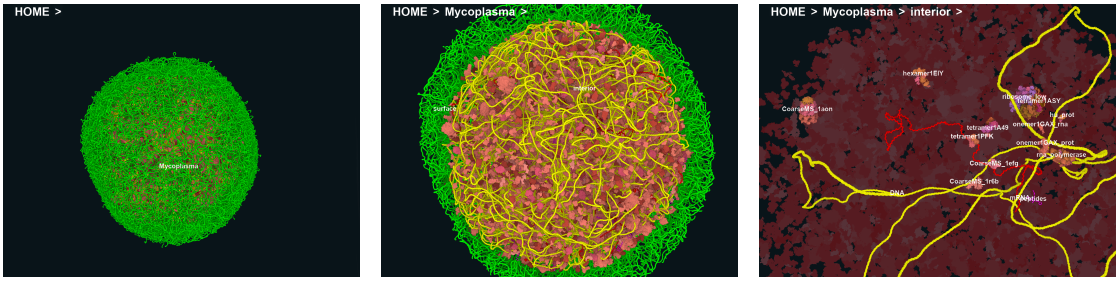
In the first model, an HIV particle is surrounded by blood plasma proteins, making it initially invisible. The interaction with HyperLabels allows us to reveal the true composition of the blood plasma and show that the virus is submerged in it. We show a

[1] Available on Vimeo: `https://vimeo.com/391959923`



(a) (b) (c)

Figure 5.8: 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.



(a) (b) (c)

Figure 5.9: 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.

tour through the composition of the virus in Figure 5.8a–(c), step-by-step revealing more internal structures of the virus.

The mycoplasma shown in Figure 5.9 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 5.9a–(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 5.8). 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 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.

5.5 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 [HMK+20] 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 [VG05] instead of sparsification for models that do not have multiple instances.

CHAPTER 6

Automated Tours of Biological Models

This chapter is based on the following work:

David Kouřil, Ondřej Strnad, Peter Mindek, Sarkis Halladjian, Tobias Isenberg, Eduard Gröller and Ivan Viola, "Molecumentary: Scalable Narrated Documentaries Using Molecular Visualization", Nov. 2020, preprint, arXiv:2011.02418 [cs.HC].

In the previous chapters, we integrated non-spatial information for various reasons in scientific visualization. The textual labels enabled an inexperienced user to gain slightly more insight into the molecular model's components. There is, however, much more information to communicate about these models to the audience.

In the method introduced in this chapter, we utilize text-to-speech technology to generate voiceover commentary. The technology that enables to turn the written word into speech has in the past years matured to the point where people might not even realize that the voice is machine-generated. We see this technology analogous to the shift from offline, pre-rendered graphics to real-time interactive visuals. With the voiceover, it has also been the practice to pre-record the commentary. With text-to-speech, the voiceover can be generated on-the-fly.

The technique for generating automated tours of biological models falls into the automative navigation category. It alleviates the user from the responsibility to control the exploration. As such, the resulting system is suitable for venues where user interaction is not possible. This work represents a step into putting data visualization to venues that interface with the general public.

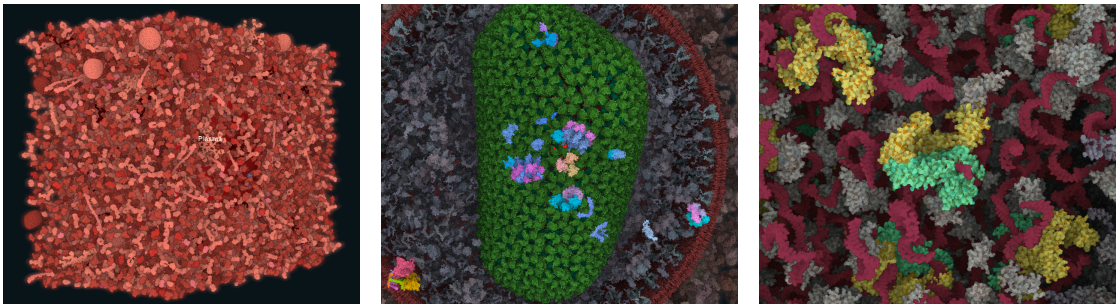
6.1 Introduction

Scientific visualization has been helping researchers to make sense of their data. Visualization today also contributes to another, increasingly important, part of science: science outreach [Var14]. A growing number of researchers now focuses on communicating the current state-of-the-art of life sciences not only to students and stakeholders, but also to the general population. Moreover, while many visualization techniques for biology have been developed, they mostly focus on transforming raw data into purely visual representations. A major issue is that, in most cases, the final image is incomprehensible to non-experts without some sort of guidance and description. Learning is possible only at specific locations where domain-expert guidance is available, e. g., schools, museums, or science centers.

Visualization used as a tool to gain insights into scientific data only works if the user is familiar with the concepts of the particular field. Without such domain knowledge, the visual representation remains a pretty image. A plethora of written materials exists in the life sciences (e. g., textbooks, online educational sites) with detailed information about the studied topic. In this medium, however, the visual and spatial characteristics of the matter are disconnected from the written explanations. Consequently, a new way of learning is becoming ubiquitous and preferred by students of life sciences today [DCM14]. Scientific concepts are often presented using computer-generated animations and uploaded to sites such as Youtube or Vimeo. These videos communicate a topic in an engaging way by leveraging storytelling techniques developed by the animation industry over decades. A verbal narration is often an essential part that contributes to the explanatory value of such material.

Yet, pre-rendered computer animations are significantly different from interactive 3D visualizations. Mainly, a computer animation undergoes a production pipeline and often cannot easily be changed after it is published, e. g., according to new scientific findings. In contrast, an interactive 3D visualization that is rendered in real-time can provide visuals immediately on demand. Developing the visuals based on real-world data makes them flexible and ready for future extension. These aspects make 3D visualization a suitable candidate for science communication, as exemplified by its application in astronomy communication [BAC+20]. The existing cases of applying visualization in science communication underline the need for incorporating explanation and guidance for public dissemination. We pose the following research question: *How can explanatory information about the function and role of individual subparts be integrated into a 3D visualization while leveraging the benefits provided by interactivity?*

We address this question with a method that elevates 3D scientific visualization into a scientific documentary (e. g., Figure 6.1). The explanatory information is integrated through verbal annotation using the auditory channel. We couple verbal annotations (i. e., the commentary) with an automatic fly-through of a structural model, providing visuals relevant to the commentary. The annotation communicates the roles and functions of the building blocks of the model, resulting in a virtual guided tour of the particular



(a) (b) (c)

Figure 6.1: In the tour of the HIV in blood plasma model we for example visit the capsid (b) which contains the genetic information of the virus. Besides the RNA, the capsid contains several important proteins, such as Reverse Transcriptase (c).

model. The result resembles a manually authored scientific documentary. Our method is completely data-driven based on the structural 3D model. We describe methods for generating fly-throughs of the model, using the hierarchical organization and the functional relationships between its components. Furthermore, we produce the verbal annotations with text-to-speech technology, which allows us to leverage content written by domain experts over many years. This makes our method scalable and suitable for life science communication, where it is highly likely to incorporate new knowledge in the future.

To realize this novel method of using real-time scientific visualization for science communication, we contribute:

- the conceptual *Scalable Documentary* framework which comprises real-time methods for producing scientific documentaries in a scalable and future-proof way;
- *Molecumentaries* as an exemplary application of the scalable documentary concept using multi-scale, multi-instance, and dense 3D molecular models;
- an automated method for *Story-Graph Foraging*, i. e., gathering descriptional information about the model components and constructing the story structure from these descriptions; and
- a method for *Real-Time Narrative Synthesis*, which interactively plans a traversal of the story graph, manages automatic cinematic camera animations, and ensures that a corresponding verbal commentary is provided with the visuals.

6.2 Scalable Documentary: Overview

To address needs in science communication, we propose the concept of *scalable documentaries* (Figure 6.2), i. e., a conceptual framework in which we use interactive visualization

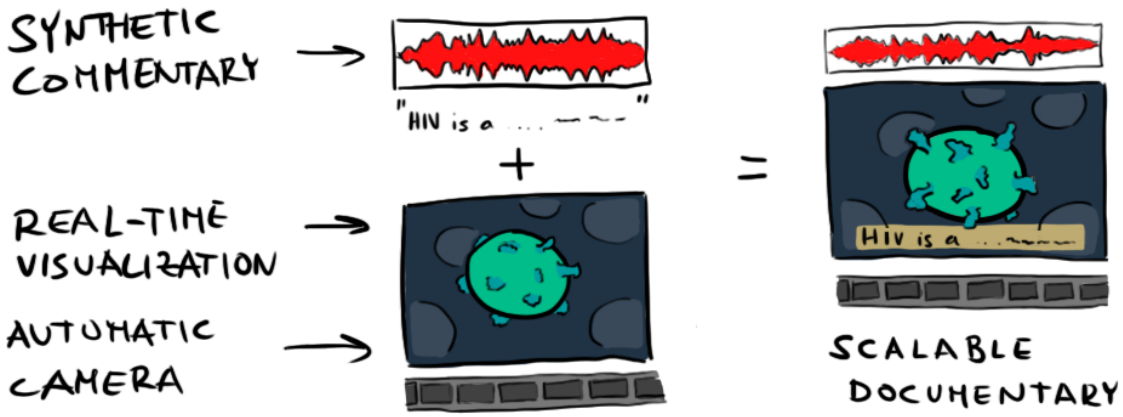


Figure 6.2: The Scalable Documentary concept: We provide visuals as a real-time visualization and couple it with an automatic camera traversing the 3D scene. We augment the fly-through with a synthetic commentary that we generate on-demand, as opposed to using a pre-recorded voice-over.

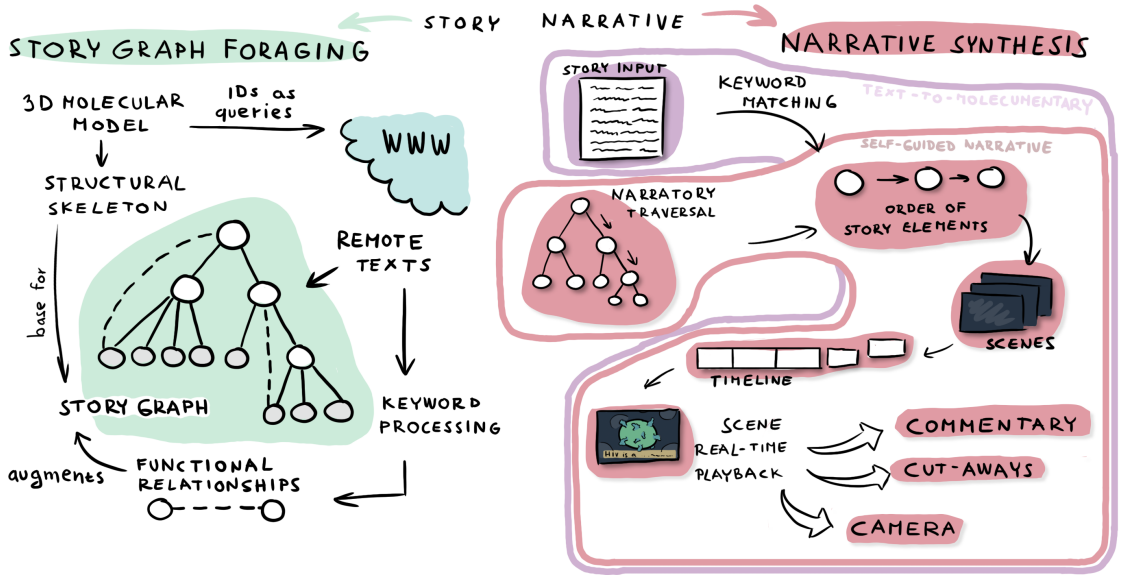


Figure 6.3: Overview of the framework for generating molecumentaries. In describing the technical contributions we follow the distinction between a story (static representation of story elements) and a narrative (dynamic arrangement of these story elements). Story-graph foraging provides a scalable approach for compiling information about the model which can be used for storytelling, while the real-time narrative synthesis encompasses solutions for turning the story graph into a specific narrative at runtime.

as a medium of science communication. We are inspired by scientific documentary movies, which explain concepts by combining computer animations and voice-over commentaries. As the name implies, we place emphasis on scalability, i. e., the ability to adapt to future inputs. Our framework rests on three basic components: the use of *real-time visualization*

instead of pre-rendered animations, an *automatic camera* that procedurally traverses and showcases the 3D scene, and the coupling of visuals with a *synthetic commentary*.

Real-Time Visualization: Real-time visualization based on actual data, as opposed to off-line rendering, allows us to introduce a high degree of interactivity. Changes to the camera position and orientation, scene lighting, and animations are immediately reflected in the visual output. The scene can also be dynamically modified to emphasize specific objects that are initially not visible due to occlusion.

Automatic Camera: Due to our use of interactive graphics we no longer can or have to record camera movements in advance. While it would be compelling to give users full control over what they wish to interactively explore, in complex multi-scale 3D environments such as in biology (e. g., a cell's inner structure) they may easily get overwhelmed and loose orientation. This issue applies particularly to our target audience of lay-persons with only a limited domain knowledge. We thus instead rely on an algorithm-controlled camera and turn the visualization user into a viewer.

Synthetic Commentary: A verbal commentary is an important part of a traditional documentary. The usual approach of pre-recording commentaries, however, does not fit into the concept of scalable documentaries. Flexibility is needed in this context, new knowledge is likely to be discovered and will need to be incorporated in the future. We thus use a procedural approach to provide the voice-over. First, we use text content written previously by domain experts. Second, we employ text-to-speech functionality to turn these textual descriptions into a verbal representation. As a result, we imitate a human commentator in a scalable way. Furthermore, this approach is, in general, language-agnostic: Texts can be queried in any given language and we can then use an appropriate speech synthesis engine.

We envision our scalable documentary concept to be applicable in several scientific domains. For the remainder of this article, however, we demonstrate it in the context of meso-scale molecular models. This exemplary case scenario represents a situation where state-of-the-art visualization methods can produce astonishing imagery. The visuals themselves are, however, mostly incomprehensible to people untrained in the domain. As a proof-of-concept we produce a scalable documentary movie that integrates additional domain knowledge and provides explanation to novices.

The involved biological models are represented and visualized on a molecular level. They exhibit several specific characteristics that we need to consider in the documentary production. First, they comprise multiple scales, exhibiting structures on scales ranging from individual atoms (approx. 0.1 nm) up to a level of a whole virus (120 nm) or even a whole cell (approx. 10 µm). Second, they rely on multiple instancing, i. e., the components that build up the model are present in large quantities, which also results in a dense packing. This density can lead to a cluttered image and we thus need to incorporate cut-aways to show the inner composition of the model. Moreover, the models are constructed through a specific technique [JAAA+15, JGA+14] with an inherent

labeling, i. e., identifiers of their components. We use these universal object identifiers in the documentary synthesis to reach an even greater level of scalability.

Before we explain the technical details of how we generate a molecumentary, we define two terms that are often used interchangeably—a *story* and a *narrative*. For the purpose of our method, we differentiate between these two.[1] We consider a *story* to be the overall architecture of story elements, e. g., events, actors, and their relationships. In contrast, we regard a *narrative* as a sequence of these story elements presented in a certain order. Different narratives of the same story can be built by changing the order of story elements. We organize the technical description of our framework along this distinction between a story and a narrative, as illustrated in Figure 6.3.

In Section 6.3, we explain how we organize a story in a data structure called a *story graph*. A story graph is often utilized in interactive storytelling [RY06]. In our case, it holds all the model elements, their relationships, and meta-information regarding the biological model. Creating such a story structure manually is tedious and in the context of a molecumentary would require the involvement of a domain expert. We present *Story Graph Foraging* as an automatic method for constructing the story graph. In Story Graph Foraging we fetch descriptions about the model components from both local and remote sources, and then extract relations between the components from these textual descriptions.

In Section 6.4, we describe how we generate the actual molecumentary. We use the story graph to generate an on-the-fly narrative, i. e., we build a sequence of story elements that will be featured in the molecumentary. Furthermore, in the narrative generation we use the descriptions stored in each story element to synthesize an on-demand commentary, using text-to-speech functionality. With automated camera animations and occlusion management we execute scenes that communicate the subcomponents of the model. We determine the order of the model elements shown in the documentary in one of two ways. In the first case, the documentary is self-guided, i. e., an algorithmic approach determines in which sequence the hierarchical structure of the model is explored. In the second case, which we call *text-to-molecumentary*, we generate visuals that follow a storyline supplied as written-text input. Moreover, the visuals can react to changes in the text directly, so the whole system can be used in real-time. The user can textually compose the story and immediately see its impact.

Our proposed concept facilitates scalable science communication. By automatizing a large portion of the scientific movie production pipeline, we are able to immediately reflect new knowledge, e. g., new research results, into the science communication medium such as scientific movie, interactive learning tool, or museum installation. We note, however, while we do deal with the terms story and narrative, that we do not attempt to solve the problem of generative storytelling. We rely on texts written by various writers, but

[1] We base our terminology on Olaf Bryan Wielk's, a storytelling theoretician: https://www.beemgee.com/blog/story-vs-narrative/

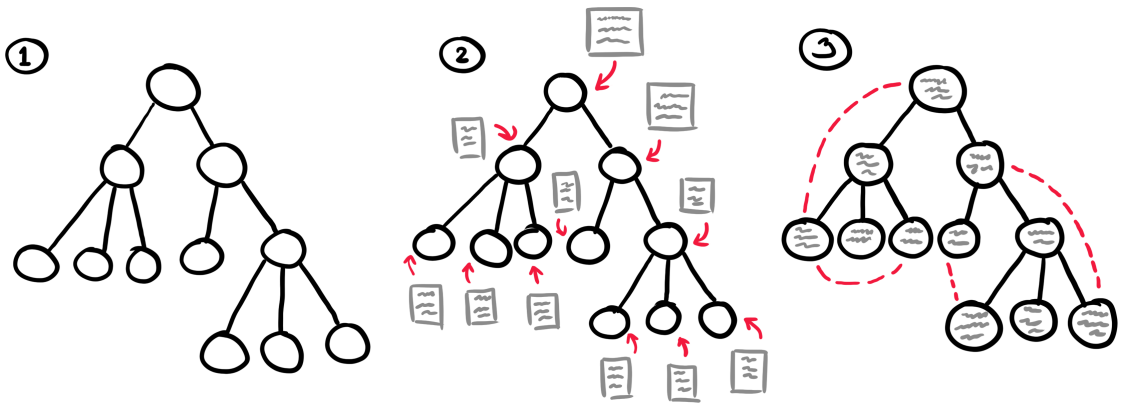


Figure 6.4: The three steps in story graph foraging. First, we build the structural skeleton. Second, we associate textual descriptions with individual nodes. Finally, we add functional relationships.

essentially consider these texts as "black boxes." We do not extract meaning and do not aim at producing a story that is creative and/or stylistically correct.

6.3 Story-Graph Foraging

At the core of our method lies the *Story Graph*, which contains data needed to build stories about a biological model. The story graph is composed of *type nodes* and *relationships edges*. Each of the nodes represents a type of a biological structure featured in the model and contains a set of descriptions detailing its role. More than one edge is allowed between two nodes, which turns the story graph a multigraph. The edges represent relationships between the structural elements. These relationships can have several types. In our work, we specifically recognize two cases: structural relationships and functional relationships. Structural relationships represent spatial and hierarchical relations of the subcomponents (e. g., *blood plasma contains the protein Albumin*). Functional relationships relate structures that are involved in a certain biological function, i. e., they interact or are related. Based on the two edge types, the story graph can be decomposed into a directed acyclic graph, which models the structural relationships (we later refer to this as the "skeleton" of the story graph), and a general multigraph, which contains the functional relationships.

In the rest of this section, we describe our method for building the story graph by *foraging*. We use the term foraging rather than construction to express the flexibility and liveliness of this process. The story graph is not only constructed once with a specific, correct result as a goal. It rather is a continuous process that can achieve different results, depending on the case and situation. This reflects the volatility of the subject matter, with new knowledge coming in, new repositories becoming available, and the large number of stories that can be told in this context. We perform story graph foraging in several steps (see Figure 6.4), each improving the resulting generated narrative.

6.3.1 Step 1: Structural Skeleton Foraging

In the first step, we organize the structural elements of the model into a basic skeleton of a story graph. We establish the structural relationships between biological structures which define the structural organization of the model. For example, for proteins building up a certain higher-level composite object, then we consider these two connected via a parent-child relationship. Such relationship is then represented by a structural relationship edge in the story graph.

We create a node for every molecular type in the model. Using the structural model, we first obtain the base skeleton for the story graph and use it to build a tree skeleton from its hierarchical organization. We then represent higher-level structures (parent objects) by their own nodes in the story graph. The names of individual components are the only descriptive information that we can relay to the viewer at this point. They can be shown, e. g., as textual labels. Furthermore, a simple narrative can be synthesized using even with a story graph just containing this structural "skeleton." However, the output would be rather rudimentary, essentially communicating the structural organization of the biological model (e. g., "Structure X contains components Y, Z, and W. Let us look at Y first.").

6.3.2 Step 2: Type Node Descriptions Foraging

We can improve the resulting narrative by incorporating descriptions about the individual structure types, which explain the role of the associated structure in the biological model. Gathering these descriptions represents the second step of story graph foraging. At the lowest level, the textual labels from the structural skeleton can be extended with additional or alternative *names* of the structure, such as some form of identification (e. g., PDBID in case of proteins). While this minimal annotation facilitates identifying the structure and distinguishing it from others, it does not provide any insight into its function. A higher-level description of the function can thus be established by expanding this annotation to the level of *one short sentence* or, of course, to *longer descriptions* with more detailed explanations. In both cases, the expressiveness and possibility of clearly communicating the message largely depends on the skill of the writer.

There are several options for getting these descriptions. First, some descriptive texts can be manually written and **supplied locally** along with the structural model. We present these text snippets with the highest priority since they are specifically created to describe the given structure. However, they might only express one level of detail and are not scalable since they have to be prepared for every element. In case no such information is provided, we thus use an alternative way of gathering descriptions. We take the standardized names of biological structures (i. e., Albumin) as keywords for searching in **remote, online repositories**.

Publicly accessible databases (such as Wikipedia) contain a large amount of interesting and relevant information written by domain experts. We take advantage of web APIs and use the name of the queried structure as a search keyword and fetch the structural

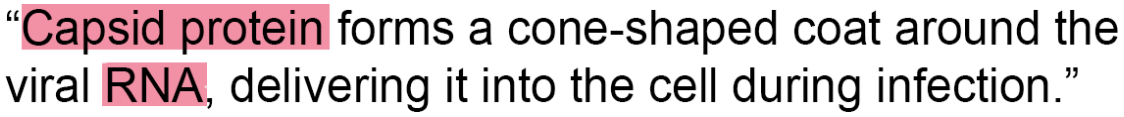


Figure 6.5: A sample textual description in which a functional relationship has been extracted. Through keyword detection the fact that the capsid protein forms a structure protecting the RNA is established. Such a functional relationship is added to the story graph as an edge.

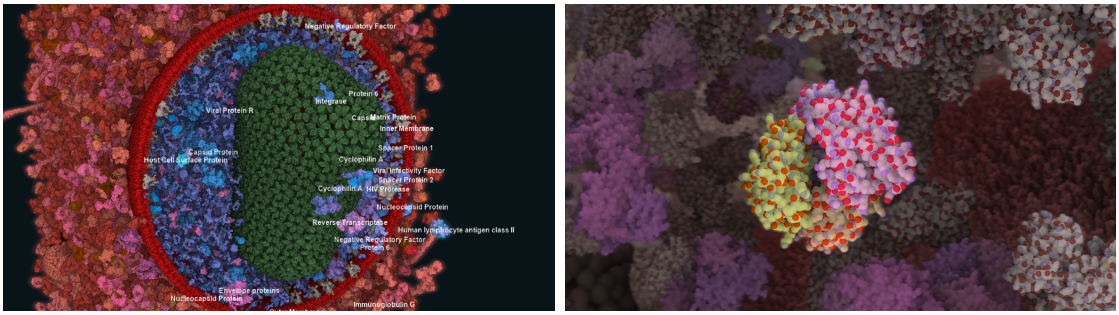
description as a response. We target short, high-level descriptions which describe the searched term in a few sentences. In our case, we use so-called extracts available in responses from the Wikipedia API which typically contain the first paragraph of the Wikipedia page for the described topic. This process can be done in real-time and we do not need to pre-fetch the descriptions for the whole model before the narration starts, saving memory. One of the biggest benefits of the real-time fetching construction is that we can scale this approach to any size of the model, provided that the model is reasonably annotated. Also, by fetching the data online in multi-lingual databases such as Wikipedia, we can query information in several languages. Furthermore, these days very powerful translation engines are becoming available and their APIs can be used as a component in the descriptions foraging pipeline. The drawback of this approach is that if the element is annotated by a generally known word that is typically used with different meanings in several domains (e. g., “plasma”) the results may not be relevant. We accept this trade-off as it is easier to modify a label to be more specific than to write a reasonable paragraph of text. A label can also contain a name for which querying does not produce text from the remote source. In this case we fallback to the structural commentary we explain in Section 6.4.1.

When descriptive texts are incorporated in the narrative synthesis, the result is a much more natural sounding documentary. The virtual narrator provides explanations to the viewer and the viewer learns about the structures visible on the screen and their functionalities.

6.3.3 Step 3: Functional Relationship Edges Foraging

So far the order of explanation is only driven by the structural organization. To relate structures that are associated not because of their proximity in the hierarchy but rather because how they interact, we need to add functional relationships to the story graph—the final step of story graph foraging.

We could establish these functional links by using data about the metabolic exchanges between structural components of the modeled organism, i. e., the metabolic pathways. Integrating such data, however, would require the intervention from a domain expert. We thus use another—text-based—approach to extract functional relationships, as illustrated in Figure 6.5. We get the names of all substructures in the model from the story graph skeleton and accumulate them into a keywords list. We then process the user-authored



(a) (b)

Figure 6.6: Overview scene (a) communicates the composition of an object while a focus scene (b) describes its function. Transition scene type is defined to switch focus between different objects and connect the other two types of scenes in the narrative.

or downloaded texts from Section 6.3.2, split them into sentences, and search these for keywords they may contain. When we detect any keyword in a sentence, we establish functional relationship edges between structures associated with these keywords. We then consider these functional neighbors when the story graph traversal method decides on which nodes shall be covered in the synthesized narrative.

6.4 Narrative Synthesis

After we created the story graph with both structural and functional information, we prepare the story for the narrative synthesis. Below we first describe the general approach for producing a specific narrative, i. e., the story elements presented in a sequence. Next, we demonstrate two scenarios of molecumentary synthesis. In one scenario we decide what is shown solely based on our story graph traversal algorithm. In the other scenario we use an human-authored, textual narrative and employ our molecumentary synthesis process to produce accompanying visuals.

6.4.1 Timeline and Scenes

We represent a specific narrative in a *timeline* data structure. The timeline is composed of a sequence of *scenes*, where each scene contains both visual and audio pieces of individual parts of the molecumentary. The timeline works as a queue—scenes are added (pushed) to the back and removed (popped) from the front, implementing the first-in first-out approach.

We use scenes of three types: focus, overview, and transition. A *focus scene* (Figure 6.6b) is the central building block of our narrative: it shows details about one structure type. We move the camera to close in on the selected instance, then use subtle rotation animations to provide parallax, and give a detailed description of the function and responsibility of the focused object inside the modeled organism.

The issue with only using focus scenes in the molecumentary is that the object in focus is shown as a whole and the viewer does not get a good idea of its internal composition. This is particularly problematic for composite objects in the hierarchical model. Therefore, we incorporate a second scene type: overview scenes. An *overview scene* (Figure 6.6a) shows all building blocks of a certain model part. The aim is to communicate the structural composition of the object. We adjust the cut-away settings of the visualization to showcase the building blocks. We highlight representative instances for each subcomponent and place the camera at such a position that shows all of them. The accompanying commentary verbalizes these components and establishes their relationships with the current object in focus.

To be able to meaningfully switch between the various focus and overview scenes, we need animated transitions that communicate a shift of emphasis. We use *transition scenes* bridging the other two types of scenes. They mostly function as connective material between the individual overview and focus scenes and provide context for a fly-through. Transition scenes usually contain significant camera transitions and changes in the visualization. The verbal commentary in these scenes then provides additional guidance and comments the transitions that happen.

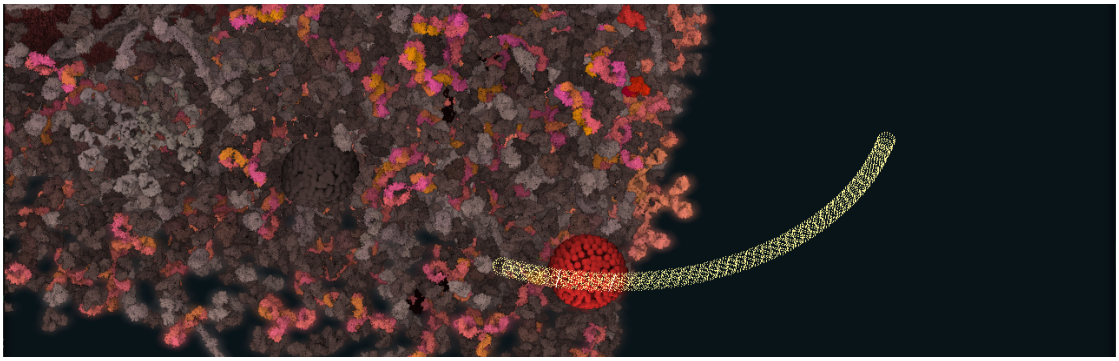
We now describe the three processes—camera animation, occlusion management, and voiceover—that we used in implementing the scenes in the molecumentary synthesis.

Camera Animation

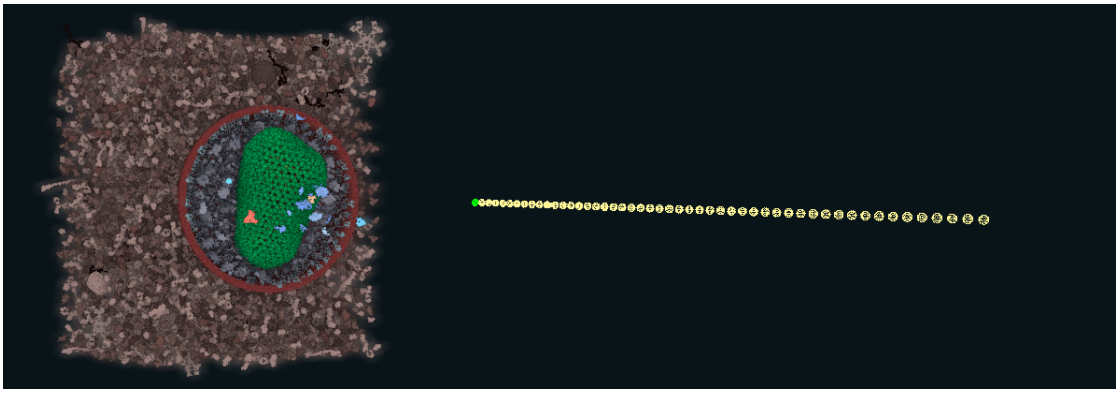
Camera movement plays an important role in conveying the multi-scale model, along with its many subcomponents. We primarily use three movement types in producing the molecumentary. These are *anchored orbiting*, *direct flying*, and *curved transition*, illustrated in Figure 6.7. Many more camera movements can be incorporated and developed for future applications. Here, we describe our basic camera language sufficient to be used in our prototypical implementation.

To be able to generate the camera animations we need to know the position and shape of the structures in focus in each scene. We use a bounding sphere as an approximation of the target object shape and size. A bounding sphere can be quickly computed in real-time and in many cases in our application scenario (molecular models) it approximates the shape sufficiently well. The camera animations are defined between targets that are specified by two attributes each: world position and a radius of the bounding sphere.

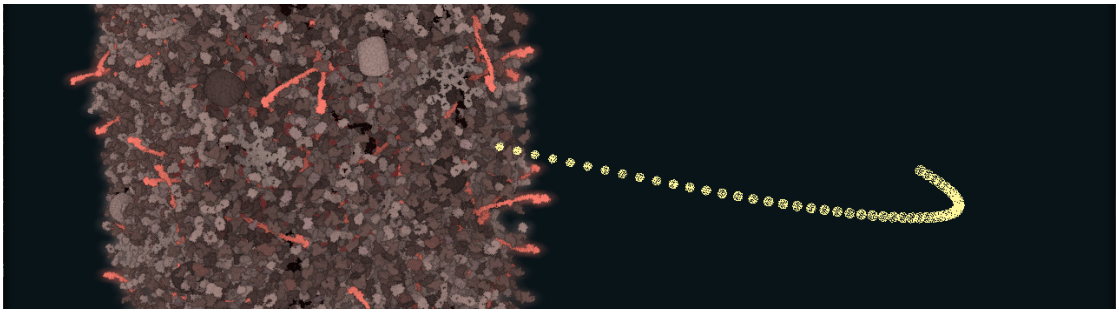
Anchored orbiting refers to a slow movement of the camera rotating around a specific object instance while keeping the camera oriented toward the center of the instance. Anchored orbiting achieves two goals: it provides 3D motion parallax and gives an impression of the local neighborhood. It thus contextualizes the focused instance in 3D space and shows neighboring structures. We use anchored orbiting in focus and overview scenes. The orbiting direction (clockwise or counterclockwise) is decided randomly in each scene.



(a)



(b)



(c)

Figure 6.7: Illustration of the three camera animation types used in a molecumentary synthesis: anchored orbiting (a), direct flying (b), and curved transition (c).

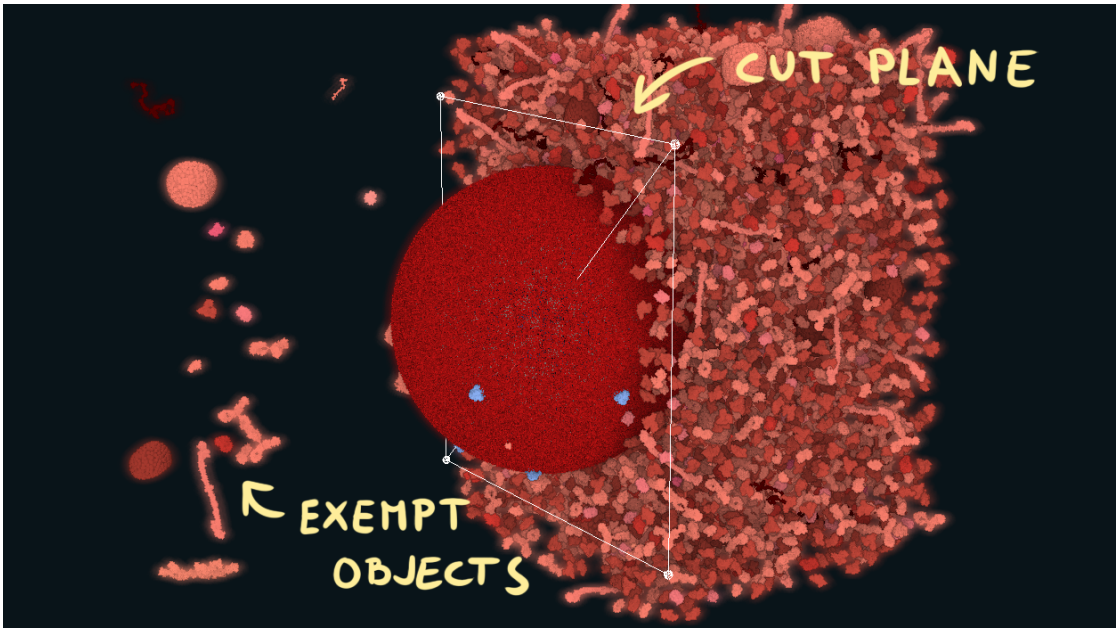


Figure 6.8: Traveling cutting plane: We remove all objects—except a selected subset—that lie between the cutting plane and the camera position to reveal inside components of the model. We implemented the cutting based on the world-space position of the scene objects. We determine the objects that are exempt from the cutting based on the scene type and the type of object that is currently shown in the molecumentary.

For a continuous narrative, however, we also need to transition between two focus instances for which we use *direct flying*. We animate the camera along a straight line, with its orientation fixed. This movement type is suitable for cases where the two instances (initial and target) are visible from the initial camera viewpoint. If the target position is outside of the viewing frustum, direct flying can be suboptimal in communicating the spatial relation between the two objects.

Therefore, we introduce the third movement type: *curved path animation*. In this animation type we zoom the camera slightly out of the initial focus position, providing context of its surroundings, and then travel toward the target focus position on a curved path. We use a normal Bézier curve defined by three points, but any curve type can be used.

Occlusion Management

Biological models are densely packed with molecules, which results in occlusion of most of the interesting structures, e. g., in the inside of a virus. Occlusion management is required to showcase all relevant parts of the model properly.

The occlusion management method we use in Molecumentaries is based on our previous work [KIK+20], where we *sparsified* the dense molecular scene.

We employ a *traveling cutting plane* approach (see Figure 6.8). We define a cutting plane in the scene and do not render any object that lies between the cutting plane and the camera. This approach has in the past been also called selective clipping [Lor93] or selective cutting [TBH+93]. Birkeland et al. [BBBV12] feature an interesting extension of this idea for volumetric models, where they define the cutting plane as an elastic membrane that conforms to structures in its proximity. As a result, structures are not strictly cut by the cutting plane, leading to a more illustrative effect. Similarly, we also exclude certain instances (or types) from being cut away. This allows us to highlight the selected objects as well as convey the impression of the absolute number of these objects in the model. The cutting plane *travels*, i. e., we animate it and the set of objects we always show throughout the molecumentary to successively reveal objects that are being verbally described. We perform the animated transitions in the *transition scenes*. We then determine the objects exempt from removal based on the type of the scene that follows the transition.

For a *focus scene* we shift attention to one (sub-)structure type. To emphasize this focus type, we exempt all its instances from being cut for the duration of the scene to communicate their number in the model. We then re-position the cutting plane to the center of a selected representative instance. We select the instance closest to the camera as the representative and orient the cutting plane to be parallel to the viewing plane at the moment the object comes into focus. We thus set the normal vector of the plane to be the same as the camera's initial back vector. In an *overview scene* we communicate inner composition of a structure. In the transition scene that leads up to it we thus create a view that shows the inside. We do so by fetching the structural components (child nodes) of the focus structure and, for each of these child nodes, pick a representative instance and exempt it from the cutting. We then place the cutting plane at the position of the representative that is furthest away from the camera such that none of the representatives is occluded by instances kept in the scene.

We purposefully used the traveling cutting plane as a world-space technique that culls instances, rather than image blending effects. The fading in and out of alpha blending resembles a "cut" in movie making, which would make it less apparent that changes in the scene communicate an opening up of the model, as opposed to a change of the scene altogether. Also, while we incorporate only one cutting plane in our design, we envision that using multiple planes would be possible. However, keeping track of the cutting planes to ensure that an object selected in the future will be visible, in our opinion, outweighs the potential benefits.

Verbal Commentary

We realize the verbal commentary with text-to-speech tools. We generate the commentary in textual form and later synthesize audio using an artificial voice. We employ three types of commentaries: structural, descriptional, and navigational.

We use *structural commentary* in overview scenes to describe the structural composition of

certain composite objects. In structural commentary we explain which sub-objects make up the described object. An example of structural commentary is *"Blood plasma consists of Hemoglobin and Heparin and others."* We construct the commentary procedurally based on the hierarchical object composition. To make the generation process scalable, we use sentence templates. Because the structural commentary relies on a hierarchy, the sentences typically contain word formations such as "consists of . . . ," "belongs to . . . ," and similar. We further define variables that can be used in the templates. We replace these variables in real-time by respective values based on the current story graph traversal. The variables are: *$name*, *$siblings*, *$children*, *$parent*, *$previous*. *$name* denotes the element on which the story currently focuses. The variables *$siblings*, *$children*, *$parent* contain hierarchical information related to the current node *$name*. The variable *$previous* points to the node which has been in focus just before *$name*. In the above example, the template we used was *"$name consists of $children"*. In large hierarchies, *$children* and *$siblings* can contain tens or hundreds of nodes—too many to list all in the commentary. Therefore, we randomly select a subset of them (we use three) to keep the sentence short. Also, the commentary likely changes in case the virtual tour returns back to the current node and the structural commentary is used again.

We employ a *descriptive commentary* in focus scenes. It provides the explanatory information about the individual components of the model. We use the previously described contents (Section 6.3.2) of the story graph nodes to synthesize its text to describe the objects' functions and significance in the model. We use pre-defined texts with a higher priority than ones fetched from online sources. We currently consider these texts as black boxes, so their expressiveness depends on their authors and we use them as is.

We use a *navigational commentary* in transition scenes. Its purpose is to contextualize what happens in the transition scene and connect the overall narration. We synthesize the sentences using the same templating as we used in the structural commentary, but with a different set of templates (e. g., *"After focusing on $previous we can see $name."*), from which we select random entries.

We also display textual labels in the scene [KIK+20] to connect the verbal narration with the shown structures and to help viewers to differentiate the mentioned objects. We dynamically place labels that name the structures on those representative instances that are relevant to the current scene.

6.4.2 Self-Guided Narrative

Given the general concept for molecumentary synthesis, we now present two variants of this scalable documentary application. Here, we first showcase a self-guided molecumentary, i. e., we do not use input that would inform the narrative to be shown. Instead, we automatically create the flythrough based on the organization of the model and a specific "narratory traversal" story graph exploration algorithm.

Narratory Traversal

In deciding the order of story nodes in the documentary, we deal with traversing the story graph structure. The story graph represents the hierarchical organization of the model and we wish to communicate this organization to the viewer. In the context of the molecumentary synthesis we aim to replicate the look and feeling of a scientific movie. For such a purpose, the traditional methods of traversing a tree or a graph data structure do not provide the desirable engaging results. The usual algorithmic traversal approaches result in a mechanical exploration and typically violate our requirement for the final documentary to be engaging.

We thus propose the more captivating strategy of *narratory traversal*, in which we step through the graph not with the goal of systematically visiting every node, but to showcase the 3D hierarchical structure represented by the graph. Naturally, a multitude of ways exist to meet such goals. Here we describe a method that uses two interconnected data structures: *the traversal stack* and *the options pool*. A stack is a data structure often used for exploring trees and graphs, and we use it to contain the nodes of the story graph. The pool contains the options for next objects to feature in the documentary. At any time, the top of the stack signifies the current node and, therefore, a level in the hierarchy. The pool structure then contains all the options (i. e., nodes) that we can access directly from the current node; i. e., (a) parent, (b) children, or (c) functionally related nodes. These nodes represent potential next targets, and we recompute the pool any time a node is pushed to or popped from the stack.

We use a stochastic approach to pick the next targets from the pool, as detailed in Algorithm 6.1. Our defining criterion is whether the potential next node has been previously shown, and if so then when. We specifically use the time of last visit to ensure that we can continuously traverse the whole model if the molecumentary is left to run for longer periods. The *Priority* function models a priority distribution among the nodes, and we define it as

$$Priority(n) = \begin{cases} P_{lower} & n \text{ is a leaf node} \\ P_{higher} & n \text{ is an inner node} \end{cases} . \tag{6.1}$$

It is also possible to incorporate manual input in the priority function, e. g., based on expert opinion for significance of a specific subset of structures in the model.

Timeline Building and Playback

The step-wise procedure for determining which nodes will be featured in the tour, however, is not yet sufficient. To produce a molecumentary we still need to turn this node sequence into a sequence of scenes that we can place onto the timeline. When a node (i. e., object type) is selected to be shown, we thus first add a transition scene from the current object in focus to the new one is put into the timeline, followed by a new focus scene for the newly selected node instance. In addition, if the selected node is a composite object (i. e.,

Algorithm 6.1: Next story node selection

```
   // options from the pool
 1 var options;
   // times of last visit
 2 var visitedTimes;
 3 min ← getMinimumValue(visitedTimes);
 4 foreach option ∈ options do
 5 |   if visitedTimes[option] = min then
 6 |   |   candidates.add(option);
 7 |   end
 8 end
 9 foreach c ∈ candidates do
10 |   priority ← Priority(c);
11 |   priorityRange ← priorityRange + priority;
12 end
13 rand ← random(0, priorityRange);
14 prioSum ← 0;
15 foreach c ∈ candidates do
16 |   priority ← Priority(c);
17 |   valA ← prioSum; valB ← prioSum + priority;
18 |   prioSum ← prioSum + priority;
19 |   if valA < rand ≤ valB then
20 |   |   next ← c;
21 |   |   break;
22 |   end
23 end
24 visitedTimes[next] ← currentTime;
25 return next;
```

Algorithm 6.2: Scene generation (self-guided narrative)

```
 1 lastScene ← timeline.last;
 2 if lastScene.type = overview then
 3 |   transitionOverviewToFocus(current, next);
 4 else
 5 |   transitionSiblings(current, next);
 6 end
 7 focus(next);
 8 if isLeaf(next) = false then
 9 |   pushToStack(next);
10 |   transitionFocusToOverview(next);
11 |   pushOverview(next);
12 end
```

an inner node in the structural skeleton of the story graph) we perform a "diving into" operation: We push the node onto the traversal stack, which leads to the pool of options being recomputed. We then generate an overview scene to convey the composition of this object, after we first added a transition scene to introduce the coming composition explanation. We detail the procedure in Algorithm 6.2.

6.4.3 Text-To-Molecumentary

Often there already exists a description of a particular model that describes the important parts and their functional behavior. Our second synthesis variant thus uses a story in a textual form as input to generate the molecumentary.

We parse the input text by sentences. In each sentence we search for the names of structures in the model and fetch the corresponding story graph node if we have a match. To prevent keywords that are frequently mentioned in the input text from being focused on and shown multiple times, we use every detected keyword only once during the whole story. Furthermore, we want to avoid many focus shifts within a short period of time. If multiple keywords are detected in a sentence, we thus use only the first keyword that has not yet been excluded as a story element.

In a second step we convert the found story graph nodes (i. e., structural types) into a series of scenes, similarly as we did it in the self-guided version. We then push these scenes to the timeline, which we later play in the same manner as explained before. The approach for generating the scenes also takes into account the hierarchical relationship between what was shown before and what shall be shown next in the molecumentary. Since the narrative in the input text can express arbitrary jumps through the hierarchy, the resulting scenes no longer communicate a node-by-node traversal of the story graph. To clearly communicate the hierarchical and encapsulation relationships, we could inject scenes showing also elements intermediate between the current and next target. However, we choose not to do so and transition directly to the next detected node because additional scenes would disrupt the narrative and cause undesired pauses in the synthetic voice-over. In our tests this worked without problems, provided that the input text was of sufficient quality. We summarize the approach in Algorithm 6.3.

6.5 Results

We developed a prototypical implementation of the molecumentary synthesis on top of the Marion library [MKS+18], which supports biology communication. Our molecular rendering uses cellVIEW's [LMAPV15] impostor approach, coupled with levels-of-detail for an efficient depiction of large molecular models.

To fetch the descriptive texts for the model elements, we could use any repository with such data. As noted above, we used Wikipedia's API to get article *extracts*: short descriptions of the keywords. The response time for such a query was ≈ 150 ms—well within the limits of a live production. We found that, in the majority of our tests, three sentences from the extracts are sufficient to describe a structure. The quality of the result, however, highly depends on the quality of the search terms, i. e., the structure identifiers in the annotated model. If the model is not well-annotated or keywords are too general, the results can be unrelated or misleading.

Our framework's component for verbalizing texts is implementation-agnostic. Since Marion is based on Qt, we are able to leverage its text-to-speech functionality. The Qt

Algorithm 6.3: Scene generation (text-to-molecumentary)

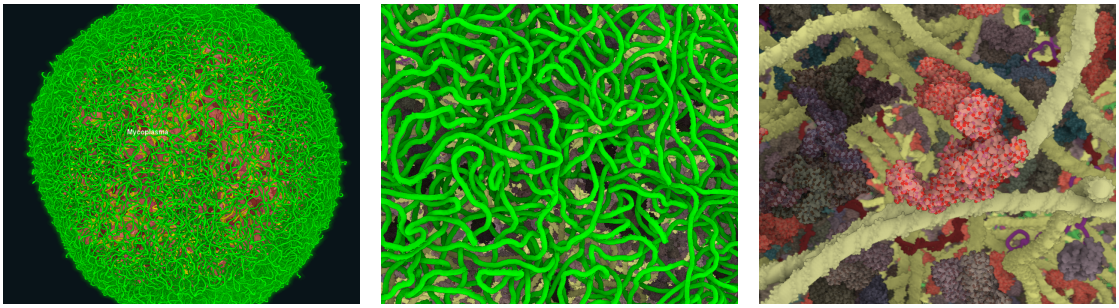
```
   // list of sentences from the text
 1 var sentences;
   // set of previously used keywords
 2 var usedKeywords;
 3 foreach s ∈ sentences do
 4 |   keywords ← identifyKeywords(s);
 5 |   keyword ← selectFirstNotIn(usedKeywords, keywords);
 6 |   current ← getType(keyword);
 7 |   if isLastSentence(s) then
 8 |   |   child ← current.root.children[0];
 9 |   |   transitionOverviewToFocus(child);
10 |   |   transitionFocusToOverview(child);
11 |   |   overviewScene(child);
12 |   else if hasChildren(current) then
13 |   |   transitionFocusToOverview(current);
14 |   |   overviewScene(current);
15 |   else
16 |   |   if previous.parent ≠ current.parent then
17 |   |   |   transitionSiblings(current.parent, current);
18 |   |   |   focusScene(current);
19 |   |   else
20 |   |   |   transitionSiblings(current, current);
21 |   |   |   focusScene(current);
22 |   |   end
23 |   end
24 |   usedKeywords.insert(keyword);
25 |   previous ← current;
26 end
```

Speech component [Com] provides an abstraction layer above text-to-speech interfaces available for several OSes, e. g., `libspeechd` for Linux or Windows' native library. As an alternative, we also interface with an online service, Google's Cloud Text-to-Speech API [Goob]. It allows us to customize several speech attributes speech and, in our experience, produces more natural sounding speech output. To support languages other than English, we use a user-defined keyword translation. These translated keywords allow us to retrieve relevant information in the target language.

We tested our approach on three molecular datasets, which we discuss next. In our supplementary video[2] we show parts from all molecumentaries produced from these datasets, which we recorded in real-time at FullHD resolution.

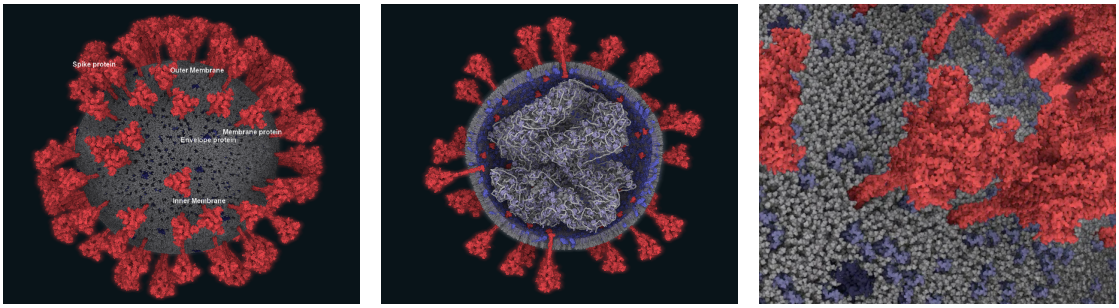
First, we use a **HIV in blood plasma** dataset (Figure 6.1) from Scripps Research. It consists of 45 protein types with ≈18,500 protein instances, 200,000 lipids, and a single RNA strand. Its hierarchy is five levels deep and the model is well annotated with descriptive names. Every molecule type has a human readable name provided by an expert, and almost all of them have a local textual description.

[2] Available on Vimeo: `https://vimeo.com/509486464`



(a) (b) (c)

Figure 6.9: The Mycoplasma model contains many more fiber instances. Throughout the virtual tour we visit both the strands visible from the outside view (e. g., peptides shown in (b)) as well as the insides of the bacteria pictured in (c).



(a) (b) (c)

Figure 6.10: The SARS-CoV-2 model shows the composition of the virus. We see its inside composition in an overview scene (b), and the very important structure—the spike protein—is then shown in detail in a focus scene (c).

For this model we asked a domain expert to provide a textual description, which we used in the text-to-molecumentary scenario. The resulting molecumentary is 2:42 minutes long and the narration is fluent as it would read by a human narrator. The transition scenes that we inserted between the focus and overview scenes for the sentences do not influence the movie's length. We also produced a self-guided movie, which we stopped after 4:33 minutes. In this time, our framework visited 11 story-graph nodes. The synthetic navigational sentences took $\approx 26\%$ of the movie's time. Using a nVidia Titan V in FullHD resolution, we are able to render both movies at at least 17 fps (in zoomed scenes), with a median of ≈ 27 fps (overview scenes).

Second, the **Mycoplasma dataset** (Figure 6.9) has also been provided by Scripps Research. The model comprises $\approx 5{,}400$ protein instances of 18 different types and RNA. Because it is a preliminary model it does not yet contain a lipid membrane. Approximately a half of the proteins is well annotated and there is no textual description provided with the model. For that reason we downloaded all needed descriptive texts

from Wikipedia and only produced a self-guided movie. We stopped this movie after visiting 11 story-graph nodes at 4:35 minutes. 24% of this time was spent on navigational sentences. The performance in FullHD resolution using nVidia Titan V was 14 fps or more.

Finally, we used the **SARS-CoV-2** dataset (Figure 6.10) provided by KAUST [NSK+20]. It consists of ≈3,200 protein instances of six different protein types, ≈180,000 lipids, and an RNA strand. The model is annotated with human readable labels, but no predefined textual description is available. We thus also retrieved all textual descriptions from Wikipedia and only produced a self-guided movie. It is 3:20 minutes long, visiting 10 story-graph nodes. The performance at FullHD resolution with the nVidia Titan V is 20 fps or more. The navigational sentences took ≈31% of the movie. With this dataset we discovered an aspect that needs improvement: Because the leaves in the hierarchy are individual RNA bases that consist of only a few atoms, the camera traversal while zooming in goes too much into the depth. The resulting view is not very attractive, and we need to address it in the future.

6.6 Discussion

To reflect on our work and validate its utility in biology communication, we showed these results to two domain experts, with 45 resp. 33 years of professional experience. They appreciated its potential in being able to showcase complex molecular models and confirmed that the many parts of these models are difficult to understand with conventional interaction methods. Both domain experts also liked the coordination of the generated speech with the visual content, commenting that the transitions are easy to follow. They also noted that our method would make a valuable tool for semi-automated content creation, provided that we add more user interaction in the creation pipeline.

The domain experts also pointed out some limitations. In particular, the final zoomed-in view does not always end up showing the molecules from a characteristic view. To solve this issue, canonical views of each structural type could be computed and used in the determination of the final camera position. Furthermore, to simplify the design of our method, we only focus on a single component at a time. One domain expert mentioned that it would be good to be able to explain two (or more) components at a time and include a commentary of their interaction. Finally, we considered only static models so far. Models from molecular dynamics simulations would present additional challenges.

In the domain of molecular and biological visualization there is a movement towards combining data from various sources and contextualizing them in a single environment [AMB+20]. The goal is to develop a pipeline that automates the whole process from data acquisition and modeling, to visualization and rendering. Our approach contributes to this effort and we consider our framework to be the initial step toward automatic interactive storytelling in the context of science communication. We can automatically integrate semantic information—fetched from online sources or provided by experts—about the composition of a molecular model. Our work is made possible by the advances

of real-time visualization. Real-time graphics, as opposed to offline rendering approaches, is being rapidly utilized in moviemaking and we believe that adopting a similar trend in visualization can fundamentally change the field of scientific outreach. Yet the field of molecular visualization still lacks sufficient standardization that would allow us to create a fully automated pipeline from observation to science communication.

Nonetheless, with our work we still contribute to the latter field of scientific outreach. While we cannot and do not intend to replace domain experts who explain specific concepts (i. e., the science communicator), with our current technology we can take advantage of the same sources that experts use, extract the key information, and deploy it on-demand to an audience at any time. As such we are able to provide visually supported scientific narratives where it was not possible to use them before, in a similar way that illustrative visualization allows us to use illustration-like visuals where we cannot afford human illustrators.

CHAPTER 7

Conclusion

The work in this thesis aims to enable people to better interact with scientific data visualization. Virtual environments built up by scientific data are often nothing like what we are usually used to from everyday life. Interacting and exploring such novel scenes, therefore, presents new challenges and requires novel technical solutions.

Our overall goal is to use real-time visualization to bring biology closer to the general public. As part of science outreach efforts, we develop methods for comprehension of the visualizations and design interactions to navigate the data. This chapter reflects on the presented technology and discusses the opportunities for future research.

7.1 Reflection

In Chapter 1, three high-level research goals were defined. This section reflects on the research conducted in the doctoral studies and discusses how it relates to the specific research goals.

Goal: *Efficiently and effectively include textual labels in complex virtual environments made up of biological models.*

Chapter 4 is primarily dedicated to fulfilling the first research goal. In this chapter, we present Labels on Levels, a novel approach to annotate dense hierarchical environments with multi-scale and multi-instance characteristics. Our labeling algorithm considers the level on which objects should be labeled and is, to the best of our knowledge, the first of its kind. We describe the entire sequence of steps that have to be processed in annotating such datasets. It works as a post-processing screen-space effect using only the structures that are typically available from the rendering of the scene itself. Therefore, Labels on Levels should be easily reproducible in other application areas as well. The labeling method is implemented as a GPU algorithm resulting in a fast processing time efficient enough for real-time use. Furthermore, we discussed the effectiveness of our approach with

domain experts who have extensive experience with biology communication, illustration, and animation.

Textual labels play a significant part in our work. The Labels on Levels technique represents a first step in incorporation an additional layer of information into the spatial data visualization. It also implements the concept of augmentive navigation, in which the visualization system provides the user with visual assistance in navigating the virtual environments.

Goal: *Enable intuitive navigation through hierarchically-organized dense 3D models.*

To simplify the navigation of a complex 3D model, we describe another mode of navigation, i. e., declarative navigation. This navigational mode, where the user does not have to control the low-level parameters to explore the model, is described in Chapter 5. Textual labels are used here, too: we leverage this verbal layer for the next target selection task. By turning the previously passive labels into interactive elements, making them *HyperLabels*, we address the problem of navigating an environment both in its spatial and hierarchical aspects.

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., [LACS08, MDLS+18]). 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 [VG05] and single-stage hierarchy exploration [PRS97], we support and emphasize 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 user's hands. With this approach, we allow users to understand—in particular for applications in biology—how parts are assembled into structural complexes. We thus allow viewers to learn both about the 3D structure, the names of the elements, and the structural composition of the complex subject matter. We envision the main application of our approach in education, yet applications are possible wherever multi-scale containment and spatial structure needs to be portrayed simultaneously.

Goal: *Automatically generate virtual tours of complex molecular structures.*

The last research goal relates to the overall high-level topic discussed by the thesis—the balance of interactivity aspects enabled by real-time visualization and a guided exploration with a specific story to tell. We address this topic by the concept of navigational modes described in Chapter 3. Creating virtual tours of complex molecular models corresponds primarily to the automative navigation category. To provide an exemplary application

of automative navigation, we propose an automatic pipeline for molecular model tours generation in Chapter 6. Based on science documentaries or narrated tours in museums, we include verbal narration together with a visual part realized by the real-time visualization. Narration, or another form of explanation, is crucial to engaging an inexperienced audience beyond the visual aspects. In the *Molecumentaries* pipeline, we utilize current text-to-speech technology to allow instant conversion of a domain expert-authored text to a verbal representation.

7.2 Outlook

While we contribute to several well-established topics in the scientific visualization field, e. g., navigation and storytelling, our work also breaks new ground in using real-time visualization methods for science dissemination. As such, it opens up many possible directions for future research, which we outline in this section.

In this thesis, we primarily focused on technical challenges regarding the navigation of multi-scale biological environments. However, our work is aimed at the interaction between the user and the data. Therefore, a logical next step would be to study and evaluate the presented techniques with the intended audience. Answering the questions of usability, intuitiveness, and effectiveness might be complicated, considering the broad user pool consisting of members of the general audience. We envision that proper evaluation with such a user group poses many open questions and elicits a formal large-scale study, possibly comprising of long-term installation of our technology at a public place, e. g., a science center.

Throughout the thesis, biological environments have been represented by static 3D models that capture the depicted structures at one particular time. Biological models are, however, in constant movement and frequently undergoing processes that change their configuration. In the future, it will be imperative that these dynamic processes are also adequately communicated. We paid close attention to the problem of multiple scales in the spatial domain. Similarly, also the temporal domain features processes that happen at different time scales. The issue of communicating the changes that have very different pace shall be investigated. We envision that real-time visualization will prove itself beneficial in this aspect, too. The ability to generate animations in real-time will provide an advantage over the time-costly pipeline of offline rendering. Scientific discoveries about the structural changes over time will be rapidly integrated into a holistic biological framework. The integrative approach, where various insights are put into a shared context, will further narrow the gap between simulation and visualization. The interactive HyperLabels should be utilized in stepping through individual stages of biological processes. That way, they will not only facilitate navigation in hierarchy and space, but also in the time domain.

Finally, we have, for the most part, worked in the traditional desktop environment, where a keyboard and a mouse set-up facilitates the interaction. Translating the presented technology into a touch interface, e. g., a mobile tablet device or an interactive touch

table should be straightforward. However, adapting the methods for other venues will require further research. We envision applications of biology communication at venues such as museums or science centers. These institutions are often more welcoming to immersive environments, e. g., Virtual Reality (VR), Augmented Reality (AR), or dome theaters.

In these scenarios, the user's impression of *immersion* is fundamental. In the example of a VR environment, we have the potential of giving the user a better sense of being surrounded by the molecular scale of life. Auxiliary to the research presented in this thesis, our group has produced a prototypical VR version of the cellVIEW system [LMAPV15], which was demonstrated in informal settings to people with varying degrees of expertise on several occasions. From this limited experience, we concur that the immersive feeling attainable in the VR environment leads to highly *engaging* experiences. The ability to submerge themselves into a virus was especially exciting for children of the early-education age. The question of how to balance the immersion, i. e., keeping as much of the dense model as possible, and the user's ability to effectively navigate the dense and multi-scale environment, i. e., traversing the model while seeing only the most immediate surroundings, should be investigated. Furthermore, it has been shown for both VR and dome theaters that certain specific design conditions must be adhered to not to cause motion sickness or feelings of anxiety.

We have scratched the surface of verbal scientific storytelling in this thesis. The story graph structure from Chapter 6 represents an initial step for automated storytelling in a biological context. We proposed a three-step procedure for filling the story graph with information that can be afterward used to synthesize a narrative. However, our prototypical implementation can be significantly extended in the future. We expect the standardization and progressing availability of information through public repositories to contribute to narratives automatically built from the available data. We envision a *living story graph* where the narrative structure can continually change and evolve. This can be either because of recent discoveries or in reaction to a specific audience. Furthermore, we believe in the potential for experience-adjusted storytelling, where the level of explanation is chosen according to the viewer's previous domain expertise. Naturally, we would like to employ different storytelling approaches for children than when explaining scientific concepts to a university student.

In our work, we essentially enable the visualization to talk to the audience, using the procedurally-generated speech. Especially in venues like a dome theater, it could be interesting to incorporate speech in the opposite direction, i. e., the audience would converse with the visualization, asking questions about specific scientific concepts. They would receive back a combination of verbal answers accompanied by visual explanations.

Bibliography

[ACS+17] Emil Axelsson, Jonathas Costa, Cláudio Silva, Carter Emmart, Alexander Bock, and Anders Ynnerman. Dynamic scene graph: Enabling scaling, positioning, and navigation in the universe. *Computer Graphics Forum*, 36(3):459–468, June 2017.

[Ada17] Eytan Adar. Banning exploration in my infovis class, 2017. `https://medium.com/@eytanadar/banning-exploration-in-my-infovis-class-9578676a4705`, accessed December 2020.

[AE05] Adel Ahmed and Peter Eades. Automatic camera path generation for graph navigation in 3D. In *Proc. APVis*, volume 45, pages 27–32, Australia, 2005. Australian Computer Society, Inc.

[AHRL+15] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Christophe Hurter, and Pourang Irani. Understanding data videos: Looking at narrative visualization through the cinematography lens. In *Proc. CHI*, pages 1459–1468, New York, 2015. ACM.

[AHS05] Kamran Ali, Knut Hartmann, and Thomas Strothotte. Label layout for interactive 3D illustrations. *Journal of the WSCG*, 13(1):1–8, 2005.

[AMB+20] Ludovic Autin, Martina Maritan, Brett A. Barbaro, Adam Gardner, Arthur J. Olson, Michel Sanner, and David S. Goodsell. Mesoscope: A web-based tool for mesoscale data integration and curation. In *Proc. MolVA*, pages 23–31, Goslar, Germany, 2020. Eurographics Assoc.

[AWM10] Hiroshi Akiba, Chaoli Wang, and Kwan-Liu Ma. AniViz: A template-based animation tool for volume visualization. *IEEE Computer Graphics and Applications*, 30(5):61–71, September/October 2010.

[Azu97] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

[BAB+17] Alexander Bock, Emil Axelsson, Karl Bladin, Costa Jonathas, Payne Gene, Territo Matthew, Joakim Kilby, Kuznetsova Masha, Carter Emmart, and Anders Ynnerman. OpenSpace: An open-source astrovisualization framework. *Journal of Open Source Software*, 2(15):281, July 2017.

[BAC+20] Alexander Bock, Emil Axelsson, Jonathas Costa, Gene Payne, Micah Acinapura, Vivian Trakinski, Carter Emmart, Cláudio Silva, Charles Hansen, and Anders Ynnerman. OpenSpace: A system for astrographics. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):633–642, Jan 2020.

[BAE+18] Alexander Bock, Emil Axelsson, Carter Emmart, Masha Kuznetsova, Charles Hansen, and Anders Ynnerman. Openspace: Changing the narrative of public dissemination in astronomical visualization from what to how. *IEEE Computer Graphics and Applications*, 38(3):44–57, 2018.

[BBBV12] Å. Birkeland, S. Bruckner, A. Brambilla, and I. Viola. Illustrative membrane clipping. *Computer Graphics Forum*, 31(3pt1):905–914, 2012.

[BCF+15] Michael A. Bekos, Sabine Cornelsen, Martin Fink, Seok-Hee Hong, Michael Kaufmann, Martin Nöllenburg, Ignaz Rutter, and Antonios Symvonis. Many-to-one boundary labeling with backbones. *Journal of Graph Algorithms and Applications*, 19(3):779–816, 2015.

[BDY06] Ken Been, Eli Daiches, and Chee Yap. Dynamic map labeling. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):773–780, 2006.

[BF08] Michael Burns and Adam Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. *ACM Transactions on Graphics*, 27(5):154:1–154:7, December 2008.

[BFH01] Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. In *Proc. UIST*, pages 101–110, New York, 2001. ACM.

[BG06] Stefan Bruckner and M. Eduard Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, September/October 2006.

[BGKG06] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and M. Eduard Gröller. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569, November/December 2006.

[BGS+13] Cynthia A. Brewer, Elaine M. Guidero, Lawrence V. Stanislawski, Barbara P. Buttenfield, and Paulo Raposo. Labeling through scale using hierarchies of thinned road networks for design of the national map of the united states. In *Proceedings of the 26th International Cartographic Conference*, pages 25–30, 2013.

[BH94] Benjamin B. Bederson and James D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proc. UIST*, pages 17–26, New York, 1994. ACM.

[BH19] Pierre Bénard and Aaron Hertzmann. Line Drawings from 3D Models: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 11(1–2):1–159, September 2019.

[BHW+07] Michael Burns, Martin Haidacher, Wolfgang Wein, Ivan Viola, and Eduard Gröller. Feature emphasis and contextual cutaways for multimodal medical visualization. In *Proc. EuroVis*, pages 275–282, Goslar, Germany, 2007. Eurographics Assoc.

[BKF+02] Nicolas Burtnyk, Azam Khan, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. StyleCam: Interactive stylized 3D navigation using integrated spatial & temporal controls. In *Proc. UIST*, pages 101–110, New York, 2002. ACM.

[BKFK06] Nicolas Burtnyk, Azam Khan, George Fitzmaurice, and Gordon Kurtenbach. ShowMotion: Camera motion based 3D design review. In *Proc. I3D*, pages 167–174, New York, 2006. ACM.

[BNC+03] Doug A. Bowman, Chris North, Jian Chen, Nicholas F. Polys, Pardha S. Pyla, and Umur Yilmaz. Information-rich virtual environments: Theory, tools, and research agenda. In *Proc. VRST*, pages 81–90, New York, 2003. ACM.

[BNPW10] Ken Been, Martin Nöllenburg, Sheung-Hung Poon, and Alexander Wolff. Optimizing active ranges for consistent dynamic map labeling. *Computational Geometry: Theory and Applications*, 43(3):312–328, 2010.

[BSB+12] Cynthia A. Brewer, Lawrence V. Stanislawski, Barbara P. Buttenfield, Paulo Raposo, Kevin A. Sparks, and Michael A. Howard. Multiscale design for the national map of the united states: Road thinning for topographic mapping. In *Proceedings of the International Symposium on Automated Cartography (AutoCarto 2012)*, pages 16–18, 2012.

[BVG10] Jean-Paul Balabanian, Ivan Viola, and Eduard Gröller. Interactive illustrative visualization of hierarchical volume data. In *Proc. Graphics Interface*, pages 137–144, Toronto, 2010. CIPS.

[BZ70] Richard E. Bellman and Lotfi A. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 17(4):B–141–164, 1970.

[CG08] Greg Cipriano and Michael Gleicher. Text scaffolds for effective surface labeling. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1675–1682, November/December 2008.

[CGM+17] Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit, and Christian Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017.

[CH06] Chris Coffin and Tobias Höllerer. Interactive perspective cut-away views for general 3D scenes. In *Proc. 3DUI*, pages 25–28, Los Alamitos, 2006. IEEE Computer Society.

[CLK14] Yao-Yi Chiang, Stefan Leyk, and Craig A. Knoblock. A survey of digital map processing techniques. *ACM Computing Surveys*, 47(1):1:1–1:44, 2014.

[CMN+05] Marc Christie, Rumesh Machap, Jean-Marie Normand, Patrick Olivier, and Jonathan Pickering. Virtual camera planning: A survey. In *Proc. Smart Graphics*, pages 40–52, Berlin, Heidelberg, 2005. Springer.

[ČmolíkB10] Ladislav Čmolík and Jiří Bittner. Layout-aware optimization for interactive labeling of 3D models. *Computers & Graphics*, 34(4):378–387, August 2010.

[CO09] Marc Christie and Patrick Olivier. Camera control in computer graphics: Models, techniques and applications. In *SIGGRAPH ASIA Courses*, pages 3:1–3:197, New York, 2009. ACM.

[Com] Qt Company. Qt speech. Web site. `https://doc.qt.io/qt-5/qtspeech-index.html`, accessed July 2020.

[CON08] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. *Computer Graphics Forum*, 27(8):2197–2218, 2008.

[CP91] James M. Clark and Allan Paivio. Dual coding theory and education. *Educational psychology review*, 3(3):149–210, 1991.

[CSC06] Carlos Correa, Deborah Silver, and Min Chen. Discontinuous displacement mapping for volume graphics. In *Proc. Volume Graphics*, pages 9–16, Goslar, Germany, 2006. Eurographics Assoc.

[CSC07] Carlos Correa, Deborah Silver, and Min Chen. Illustrative deformation for data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1320–1327, November/December 2007.

[DCM14] Craig Daly, Lauren Clunie, and Minhua Ma. From microscope to movies: 3D animations for teaching physiology. *Microscopy and Analysis*, 28(6):7–10, September/October 2014.

[DeL02] Warren L. DeLano. PyMOL: An open-source molecular graphics tool. *CCP4 Newsletter on protein crystallography*, 40(1):82–92, 2002.

[DH02] Brent M. Dennis and Christopher G. Healey. Assisted navigation for large information spaces. In *Proceedings of the Conference on Visualization '02*, VIS '02, page 419–426, USA, 2002. IEEE Computer Society.

[DVRH07] Ove Daae Lampe, Ivan Viola, Nathalie Reuter, and Helwig Hauser. Two-level approach to efficient visualization of protein dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1616–1623, 2007.

[DWE03] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Interactive cutaway illustrations. *Computer Graphics Forum*, 22(3):523–532, September 2003.

[EET07] Niklas Elmqvist and Mihail Eduard Tudoreanu. Occlusion management in immersive and desktop 3D virtual environments: Theory and evaluation. *International Journal of Virtual Reality*, 6(2):21–32, June 2007.

[Elm05] Niklas Elmqvist. Balloonprobe: Reducing occlusion in 3D using interactive space distortion. In *Proc. VRST*, pages 134–137, New York, 2005. ACM.

[ET08] Niklas Elmqvist and Philippas Tsigas. A taxonomy of 3D occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, September/October 2008.

[FB95] George W. Furnas and Benjamin B. Bederson. Space-scale diagrams: Understanding multiscale interfaces. In *Proc. CHI*, pages 234–241, New York, 1995. ACM.

[FBG+18] Katarína Furmanová, Jan Byška, Eduard M. Gröller, Ivan Viola, Jan Paleček, and Barbora Kozlíková. COZOID: Contact zone identifier for visual analysis of protein-protein interactions. *BMC Bioinformatics*, 19(1):125:1–125:17, April 2018.

[FKE13] Martin Falk, Michael Krone, and Thomas Ertl. Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. *Computer Graphics Forum*, 32(8):195–206, 2013.

[FP03] Jean-Daniel Fekete and Catherine Plaisant. Excentric labeling: Dynamic neighborhood labeling for data visualization. In *The Craft of Information Visualization*, pages 316 – 323. 2003.

[Fra02] Eric Francoeur. Cyrus levinthal, the kluge and the origins of interactive molecular graphics. *Endeavour*, 26(4):127 – 131, 2002.

[FS92] Steven Feiner and Dorée Duncan Seligmann. Cutaways and ghosting: Satisfying visibility constraints in dynamic 3D illustrations. *The Visual Computer*, 8(5–6):292–302, September 1992.

[GAHS05a] Timo Götzelmann, Kamran Ali, Knut Hartmann, and Thomas Strothotte. Adaptive labeling for illustrations. In *Proc. Pacific Graphics*, pages 64–66, 2005.

[GAHS05b] Timo Götzelmann, Kamran Ali, Knut Hartmann, and Thomas Strothotte. Form follows function: Aesthetic interactive labels. In *Proc. CAe*, pages 193–200, Goslar, Germany, 2005. Eurographics Assoc.

[Gal95] Tinsley A. Galyean. Guided navigation of virtual environments. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 103–ff, 1995.

[GFH18] David S. Goodsell, Margaret A. Franzen, and Tim Herman. From atoms to cells: Using mesoscale landscapes to construct visual narratives. *Journal of Molecular Biology*, 430(21):3954 – 3968, 2018.

[GHS06a] Timo Götzelmann, Knut Hartmann, and Thomas Strothotte. Agent-based annotation of interactive 3D visualizations. In *Proc. Smart Graphics*, pages 24–35, Berlin, Heidelberg, 2006. Springer.

[GHS06b] Timo Götzelmann, Knut Hartmann, and Thomas Strothotte. Contextual grouping of labels. In *SimVis*, pages 245–258, 2006.

[GKM+15] Sebastian Grottel, Michael Krone, Christoph Müller, Guido Reina, and Thomas Ertl. Megamol—a prototyping framework for particle-based visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(2):201–214, 2015.

[Gla01] Andrew Glassner. Interactive storytelling: People, stories, and games. In Olivier Balet, Gérard Subsol, and Patrice Torguet, editors, *Virtual Storytelling Using Virtual Reality Technologies for Storytelling*, pages 51–60, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[GLC+18] Quentin Galvane, Christophe Lino, Marc Christie, Julien Fleureau, Fabien Servant, François-Louis Tariolle, and Philippe Guillotel. Directing cinematographic drones. *ACM Transactions on Graphics*, 37(3):34:1–34:18, August 2018.

[GLG+16] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. From visual exploration to storytelling and back again. *Computer Graphics Forum*, 35(3):491–500, June 2016.

[GNR16] Andreas Gemsa, Martin Nöllenburg, and Ignaz Rutter. Consistent labeling of rotating maps. *Journal of Computational Geometry*, pages 308–331, 2016.

[Gooa] David S. Goodsell. Illustrations for public use. `http://mgl.scripps.edu/people/goodsell/illustration/public`, accessed December 2020.

[Goob] Google. Cloud text-to-speech API. `https://cloud.google.com/text-to-speech/docs/reference/rest/`, accessed July 2020.

[Goo09] David S. Goodsell. *The machinery of life*. Springer Science & Business Media, 2009.

[GP01] Nahum Gershon and Ward Page. What storytelling can do for information visualization. *Communications of the ACM*, 44(8):31–37, August 2001.

[HAS04] Knut Hartmann, Kamran Ali, and Thomas Strothotte. Floating labels: Applying dynamic potential fields for label layout. In *Proc. Smart Graphics*, pages 101–113, Berlin, Heidelberg, 2004. Springer.

[HD11] Jessica Hullman and Nicholas Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2231–2240, December 2011.

[HDS+96] William Humphrey, Andrew Dalke, Klaus Schulten, et al. VMD: Visual molecular dynamics. *Journal of molecular graphics*, 14(1):33–38, 1996.

[Her06] Angel Herraez. Biomolecules in the computer: Jmol to the rescue. *Biochemistry and Molecular Biology Education*, 34(4):255–261, 2006.

[HMC11] Wei-Hsien Hsu, Kwan-Liu Ma, and Carlos Correa. A rendering framework for multiscale views of 3D models. *ACM Transactions on Graphics*, 30(6):1–10, December 2011.

[HMK+97] Lichan Hong, Shigeru Muraki, Arie Kaufman, Dirk Bartz, and Taosong He. Virtual voyage: Interactive navigation in the human colon. In

Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97, page 27–34, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[HMK+20] Sarkis Halladjian, Haichao Miao, David Kouřil, M. Eduard Gröller, Ivan Viola, and Tobias Isenberg. Scaletrotter: Illustrative visual travels across negative scales. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), January 2020. To appear.

[HPL14] Zhi-Dong Huang, Sheung-Hung Poon, and Chun-Cheng Lin. Boundary labeling with flexible label positions. In *Proc. Algorithms and Computation*, pages 44–55, Cham, Switzerland, 2014. Springer International Publishing.

[HR07] Jeffrey Heer and George Robertson. Animated transitions in statistical data graphics. *IEEE Transaction on Visualization and Computer Graphics*, 13(6):1240–1247, November/December 2007.

[HSC+05] Justin Hensley, Thorsten Scheuermann, Greg Coombe, Montek Singh, and Anselmo Lastra. Fast summed-area table generation and its applications. *Computer Graphics Forum*, 24:547–555, 2005.

[IFH+03] Tobias Isenberg, Bert Freudenberg, Nick Halper, Stefan Schlechtweg, and Thomas Strothotte. A developer's guide to silhouette algorithms for polygonal models. *IEEE Computer Graphics and Applications*, 23(4):28–37, July/August 2003.

[Imh75] Eduard Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.

[Iwa14] Janet Iwasa. Crafting a career in molecular animation. *Molecular Biology of the Cell*, 25(19):2891–2893, 2014. PMID: 25267313.

[JAAA+15] Graham T. Johnson, Ludovic Autin, Mostafa Al-Alusi, David S. Goodsell, Michel F. Sanner, and Arthur J. Olson. cellPACK: A virtual mesoscope to model and visualize structural systems biology. *Nature Methods*, 12(1):85–91, December 2015.

[JD12] Jacek Jankowski and Stefan Decker. A dual-mode user interface for accessing 3D content on the world wide web. In *Proc. WWW*, pages 1047–1056, New York, 2012. ACM.

[JGA+14] Graham T. Johnson, David S. Goodsell, Ludovic Autin, Stefano Forli, Michel F. Sanner, and Arthur J. Olson. 3D molecular models of whole HIV-1 virions generated with cellPACK. *Faraday Discussions*, 169:23–44, September 2014.

[JH13] Jacek Jankowski and Martin Hachet. A survey of interaction techniques for interactive 3D environments. In *Eurographics State of the Art Reports*, pages 65–93, Goslar, Germany, 2013. Eurographics Assoc.

[Kar18] Ramita Karpe. A survey :On text to speech synthesis. *International Journal for Research in Applied Science and Engineering Technology*, 6(03):351–355, March 2018.

[KBUF14] Patrick Knöbelreiter, René Berndt, Torsten Ullrich, and Dieter W. Fellner. Automatic fly-through camera animations for 3D architectural repositories. In *Proc. GRAPP*, pages 335–341. IEEE, 2014.

[KČmolíkK+19] David Kouřil, Ladislav Čmolík, Barbora Kozlíková, Hsiang-Yun Wu, Graham Johnson, David S. Goodsell, Arthur Olson, M. Eduard Gröller, and Ivan Viola. Labels on levels: Labeling of multi-scale multi-instance and crowded 3D biological environments. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):977–986, January 2019.

[KIK+20] David Kouřil, Tobias Isenberg, Barbora Kozlíková, Miriah Meyer, Eduard Gröller, and Ivan Viola. HyperLabels: Browsing of dense and hierarchical molecular 3D models. *IEEE Transactions on Visualization and Computer Graphics*, 2020. To appear.

[KKF+17] Barbora Kozlíková, Michael Krone, Martin Falk, Norbert Lindow, Marc Baaden, Daniel Baum, Ivan Viola, Julius Parulek, and Hans-Christian Hege. Visualization of biomolecular structures: State of the art revisited. In *Computer Graphics Forum*, volume 36, pages 178–204. Wiley Online Library, 2017.

[KKL+16] Michael Krone, Barbora Kozlíková, Norbert Lindow, Marc Baaden, Daniel Baum, Julius Parulek, Hans-Christian Hege, and Ivan Viola. Visual analysis of biomolecular cavities: State of the art. *Computer Graphics Forum*, 35(3):527–551, 2016.

[KKS+05] Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. Hovercam: Interactive 3D navigation for proximal object inspection. In *Proc. I3D*, pages 73–80, New York, 2005. ACM.

[KM13] Robert Kosara and Jock Mackinlay. Storytelling: The next step for visualization. *IEEE Computer*, 46(5):44–50, May 2013.

[KNBP06] Regis Kopper, Tao Ni, Doug A. Bowman, and Márcio Pinho. Design and evaluation of navigation techniques for multiscale virtual environments. In *Proc. VR*, pages 175–182, Los Alamitos, 2006. IEEE Computer Society.

[Kru05] Steve Krug. *Don't Make Me Think: A Common Sense Approach to the Web.* New Riders Publishing, Thousand Oaks, CA, USA, 2nd edition, 2005.

[KSJ+14] Bum Chul Kwon, Florian Stoffel, Dominik Jäckle, Bongshin Lee, and Daniel Keim. VisJockey: Enriching data stories through orchestrated interactive visualization. In *Proc. Computation+Journalism Symp.*, New York, 2014. Brown Institute for Media Innovation.

[KSW06] Jens Krüger, Jens Schneider, and Rudiger Westermann. ClearView: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):941–948, September/October 2006.

[LACS08] Wilmot Li, Maneesh Agrawala, Brian Curless, and David Salesin. Automated generation of interactive 3D exploded view diagrams. *ACM Transactions on Graphics*, 27(3):101:1–101:7, August 2008.

[LBH12] Norbert Lindow, Daniel Baum, and Hans-Christian Hege. Interactive rendering of materials and biological structures on atomic and nanoscopic scale. *Comput. Graph. Forum*, 31(3pt4):1325–1334, June 2012.

[LCKM05] Ronghua H. Liang, Gordon J. Clapworthy, Meleagros Krokos, and Rafael Mayoral. Real-time predefined shape cutaway with parametric boundaries. In *Proc. CGIV*, pages 227–231, Los Alamitos, 2005. IEEE Computer Society.

[LCL+10] Christophe Lino, Marc Christie, Fabrice Lamarche, Guy Schofield, and Patrick Olivier. A real-time cinematography system for interactive 3D environments. In *Proc. SCA*, pages 139–148, Goslar, Germany, 2010. Eurographics Assoc.

[Le 16] Mathieu Le Muzic. *From Atoms to Cells: Interactive and Illustrative Visualization of Digitally Reproduced Lifeforms.* PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/E193-02, A-1040 Vienna, Austria, October 2016.

[LHM14] Isaac Liao, Wei-Hsien Hsu, and Kwan-Liu Ma. Storytelling via navigation: A novel approach to animation for scientific visualization. In *Proc. Smart Graphics*, pages 1–14, Cham, Switzerland, 2014. Springer.

[LHV12] Endre M. Lidal, Helwig Hauser, and Ivan Viola. Geological storytelling – Graphically exploring and communicating geological sketches. In *Proc. SBIM*, pages 11–20, Goslar, Germany, 2012. Eurographics Assoc.

[Lin10] Chun-Cheng Lin. Crossing-free many-to-one boundary labeling with hyperleaders. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 185–192, 2010.

[LKM+17] Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug A. Bowman, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, USA, Second edition, 2017.

[LKY08] Chun-Cheng Lin, Hao-Jen Kao, and Hsu-Chun Yen. Many-to-one boundary labeling. *Journal of Graph Algorithms and Applications*, 12(3):319–356, 2008.

[LMAPV15] Mathieu Le Muzic, Ludovic Autin, Julius Parulek, and Ivan Viola. cellVIEW: A tool for illustrative and multi-scale rendering of large biomolecular datasets. In *Proc. VCBM*, pages 61–70, Goslar, Germany, 2015. Eurographics Assoc.

[LMMS+16] Mathieu Le Muzic, Peter Mindek, Johannes Sorger, Ludovic Autin, David Goodsell, and Ivan Viola. Visibility equalizer: Cutaway visualization of mesoscopic biological models. *Computer Graphics Forum*, 35(3):161–170, June 2016.

[LMPSV14] Mathieu Le Muzic, Julius Parulek, Anne-Kristin Stavrum, and Ivan Viola. Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. *Computer Graphics Forum*, 33(3):141–150, 2014.

[Lor93] W. E. Lorensen. Geometric clipping using boolean textures. In *Proceedings Visualization '93*, pages 268–274, 1993.

[LRA+07] Wilmot Li, Lincoln Ritter, Maneesh Agrawala, Brian Curless, and David Salesin. Interactive cutaway illustrations of complex 3D models. *ACM Transactions on Graphics*, 26(3):31:1–31:11, August 2007.

[LRIC15] Bongshin Lee, Nathalie H. Riche, Petra Isenberg, and Sheelagh Carpendale. More than telling a story: Transforming data into visually shared stories. *IEEE Computer Graphics and Applications*, 35(5):84–90, September/October 2015.

[Lue03] David P. Luebke. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003.

[LXH+20] Can Liu, Liwenhan Xie, Yun Han, Datong Wei, and Xiaoru Yuan. AutoCaption: An approach to generate natural language description from visualization automatically. In *Proc. PacificVis*, pages 191–195, Los Alamitos, 2020. IEEE Computer Society.

[LYT12] Pablo Martinez Lerin, Daisuke Yamamoto, and Naohisa Takahashi. Making a pictorial and verbal travel trace from a GPS trace. In *Proc. W2GIS*, pages 98–115, Berlin, Heidelberg, 2012. Springer.

[MAAS15] Ali Mahdavi-Amiri, Troy Alderson, and Faramarz Samavati. A survey of digital earth. *Computers & Graphics*, 53:95 – 117, 2015.

[MCR90] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson. Rapid controlled movement through a virtual 3D workspace. *ACM SIGGRAPH Computer Graphics*, 24(4):171–176, August 1990.

[MD06] Stefan Maass and Jürgen Döllner. Dynamic annotation of interactive environments using object-integrated billboards. *The 14th international Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 327–334, 2006.

[MDLS+18] Haichao Miao, Elisa De Llano, Johannes Sorger, Yasaman Ahmadi, Tadija Kekic, Tobias Isenberg, M. Eduard Gröller, Ivan Barišić, and Ivan Viola. Multiscale visualization and scale-adaptive modification of DNA nanostructures. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):1014–1024, January 2018.

[MKA+19] Xavier Martinez, Michael Krone, Naif Alharbi, Alexander S. Rose, Robert S. Laramee, Sean O'Donoghue, Marc Baaden, and Matthieu Chavent. Molecular graphics: Bridging structural biologists and computer scientists. *Structure*, 27(11):1617–1623, November 2019.

[MKK+19] Haichao Miao, Tobias Klein, David Kouřil, Peter Mindek, Karsten Schatz, M. Eduard Gröller, Barbora Kozlíková, Tobias Isenberg, and Ivan Viola. Multiscale molecular visualization. *Journal of Molecular Biology*, 431(6):1049–1070, March 2019.

[MKS+18] Peter Mindek, David Kouřil, Johannes Sorger, Daniel Toloudis, Blair Lyons, Graham Johnson, M. Eduard Gröller, and Ivan Viola. Visualization multi-pipeline for communicating biology. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):883–892, January 2018.

[MLF+12] Kwan-Liu Ma, Isaac Liao, Jennifer Frazier, Helwig Hauser, and Helen Nicole Kostis. Scientific storytelling using visualization. *IEEE Computer Graphics and Applications*, 32(1):12–19, January 2012.

[MMGK09] James McCrae, Igor Mordatch, Michael Glueck, and Azam Khan. Multiscale 3D navigation. In *Proc. I3D*, pages 7–14, New York, 2009. ACM.

[Mot07] Kevin Mote. Fast point-feature label placement for dynamic visualizations. *Information Visualization*, 6(4):249–260, 2007.

[MP09] Konrad Mühler and Bernhard Preim. Automatic textual annotation for surgical planning. In *Proc. VMV*, pages 277–284. DNB, 2009.

[MR95] Tara M. Madhyastha and Daniel A. Reed. Data sonification: Do you see what I hear? *IEEE Software*, 12(2):45–56, March 1995.

[MS91] Joe Marks and Stuart Merrill Shieber. The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard Computer Science Group, 1991.

[MTB03] Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. In *Proc. Visualization*, pages 401–408, Los Alamitos, 2003. IEEE Computer Society.

[Mun14] Tamara Munzner. *Visualization analysis and design*. CRC press, 2014.

[MvV+15] Peter Mindek, Ladislav Čmolík, Ivan Viola, M. Eduard Gröller, and Stefan Bruckner. Automatized summarization of multiplayer games. In *Proc. SCCG*, pages 73–80, Bratislava, 2015. Comenius University Bratislava.

[NMD+17] Tobias Nägeli, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics*, 36(4):132:1–132:10, July 2017.

[NSK+20] Ngan Nguyen, Ondřej Strnad, Tobias Klein, Deng Luo, Ruwayda Alharbi, Peter Wonka, Martina Maritan, Peter Mindek, Ludovic Autin, David S. Goodsell, and Ivan Viola. Modeling in the time of COVID-19: Statistical and rule-based mesoscale models. arXiv preprint, 2020.

[OJP14] Steffen Oeltze-Jafra and Bernhard Preim. Survey of labeling techniques in medical visualizations. In *Proc. VCBM*, pages 199–208, Goslar, Germany, 2014. Eurographics Assoc.

[OSTG09] Thomas Oskam, Robert W. Sumner, Nils Thuerey, and Markus Gross. Visibility transition planning for dynamic camera control. In *Proc. SCA*, page 55–65, New York, 2009. ACM.

[Per05] Ken Perlin. Toward interactive narrative. In Gérard Subsol, editor, *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling*, pages 135–147, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Pet09] Matt Pettineo. Scintillating snippets: Reconstructing position from depth. `https://mynameismjp.wordpress.com/2009/03/10/reconstructing-position-from-depth/`, 2009. Accessed: 2018-06-26.

[PF93] Ken Perlin and David Fox. Pad: An alternative approach to the computer interface. In *Proc. SIGGRAPH*, pages 57–64, New York, 1993. ACM.

[PGH+04] Eric F. Pettersen, Thomas D. Goddard, Conrad Huang, Gregory Couch, Daniel M. Greenblatt, Elaine Meng, and Thomas Ferrin. UCSF Chimera——a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[PM20] Bernhard Preim and Monique Meuschke. A survey of medical animations. *Computer and Graphics*, 90:145–168, 2020.

[PR13] Renato D. Prado and Alberto B. Raposo. Real-time label visualization in massive CAD models. In *Proc. CAD/Graphics*, pages 337–344, Los Alamitos, 2013. IEEE Computer Society.

[PRS+95] Bernhard Preim, Alf Ritter, Thomas Strothotte, Tilo Pohle, David R. Forsey, and Lyn Bartram. Consistency of rendered images and their textual labels. In *Proc. CompuGraphics*, pages 201–210, Queluz, Portugal, 1995. GRASP.

[PRS97] Bernhard Preim, Andreas Raab, and Thomas Strothotte. Coherent zooming of illustrations with 3D-graphics and text. In *Proc. Graphics Interface*, pages 105–113, Toronto, 1997. CIPS.

[PS05] Sheung-Hung Poon and Chan-Su Shin. Adaptive zooming in point set labeling. In *Fundamentals of Computation Theory*, pages 233–244, 2005.

[PS18] Bernhard Preim and Patrick Saalfeld. A survey of virtual human anatomy education systems. *Computers & Graphics*, 71:132–153, April 2018.

[RBL+17] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Höllerer, and Eun Kyoung Choe. ChartAccent: Annotation for data-driven storytelling. In *Proc. PacificVis*, pages 230–239, Los Alamitos, 2017. IEEE Computer Society.

[Ric77] Frederic M. Richards. Areas, volumes, packing, and protein structure. *Annual Review of Biophysics and Bioengineering*, 6(1):151–176, 1977. PMID: 326146.

[RPRH07] Timo Ropinski, Jörg-Stefan Prassni, Jan Roters, and Klaus Hinrichs. Internal labels as shape cues for medical illustration. In *Proc. VMV*, pages 203–212. DNB, 2007.

[RT06] Guodong Rong and Tiow-Seng Tan. Jump flooding in gpu with applications to voronoi diagram and distance transform. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, pages 109–116, New York, NY, USA, 2006. ACM.

[RY06] Mark O. Riedl and R. Michael Young. From linear story generation to branching story graphs. *IEEE Computer Graphics and Applications*, 26(3):23–31, 2006.

[Sch15] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.

[SD08] Thierry Stein and Xavier Décoret. Dynamic label placement for improved interactive exploration. In *Proc. NPAR*, pages 15–21, New York, 2008. ACM.

[SF93] Dorée Duncan Seligmann and Steven Feiner. Supporting interactivity in automated 3D illustrations. In *Proc. IUI*, pages 37–44, New York, 1993. ACM.

[SGLM03] Brian Salomon, Maxim Garber, Ming C Lin, and Dinesh Manocha. Interactive navigation in complex environments using path planning. In *Proc. I3D*, pages 41–50, New York, 2003. ACM.

[SH07] Thorsten Scheuermann and Justin Hensley. Efficient histogram generation using scattering on GPUs. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 33–37. ACM, 2007.

[SH10] Edward Segel and Jeffrey Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, Nov 2010.

[Shn96] Ben Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[SHWZ14] Nadine Schwartges, Jan-Henrik Haunert, Alexander Wolff, and Dennis Zwiebler. Point labeling with sliding labels in interactive maps. In *Proceedings of the 17th AGILE Conference on Geographic Information Science*, pages 295–310, 2014.

[SJMS19] Antoni Sagristà, Stefan Jordan, Thomas Müller, and Filip Sadlo. Gaia Sky: Navigating the Gaia catalog. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1070–1079, January 2019.

[SMKH08] Terry A. Slocum, Robert B. McMaster, Fritz C. Kessler, and Hugh H. Howard. *Thematic Cartography and Geovisualization*. Pearson, 2008.

[SMR+17] Johannes Sorger, Peter Mindek, Peter Rautek, M. Eduard Gröller, Graham Johnson, and Ivan Viola. Metamorphers: Storytelling templates for illustrative animated transitions in molecular visualization. In *Proc. SCCG*, pages 27–36, New York, 2017. ACM.

[ST90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-D shapes. *SIGGRAPH Computer Graphics*, 24(4):197–206, September 1990.

[SVB17] Desai Siddhi, Jashin M. Verghese, and Desai Bhavik. Survey on various methods of text to speech synthesis. *International Journal of Computer Applications*, 165(6):26–30, May 2017.

[TBH+93] Ulf Tiede, Michael Bomans, Karl Heinz Höhne, Andreas Pommert, Martin Riemer, Thomas Schiemann, Rainer Schubert, and Werner Lierse. A computerized three-dimensional atlas of the human skull and brain. *American Journal of Neuroradiology*, 14(3):551–559, 1993.

[TCM06] Marco Tarini, Paolo Cignoni, and Claudio Montani. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1237–1244, 2006.

[TFBB14] Mikael Trellet, Nicolas Férey, Marc Baaden, and Patrick Bourdot. Content-guided navigation in multimeric molecular complexes. In *Proc. BIOSTEC*, pages 76–81, Portugal, 2014. SCITEPRESS.

[TIP05] Christian Tietjen, Tobias Isenberg, and Bernhard Preim. Combining silhouettes, shading, and volume rendering for surgery education and planning. In *Proc. EuroVis*, pages 303–310, 335, Goslar, Germany, 2005. Eurographics Assoc.

[TKGS14] Markus Tatzgern, Denis Kalkofen, Raphael Grasset, and Dieter Schmalstieg. Hedgehog labeling: View management techniques for external labels in 3d space. In *Virtual Reality (VR), 2014 iEEE*, pages 27–32. IEEE, 2014.

[TRB+18] Chao Tong, Richard Roberts, Rita Borgo, Sean Walton, Robert S. Laramee, Kodzo Wegba, Aidong Lu, Yun Wang, Huamin Qu, Qiong Luo, and Xiaojuan Ma. Storytelling and visualization: An extended survey. *Information*, 9(3):65:1–65:42, March 2018.

[TRCC01] Desney S. Tan, George G. Robertson, Mary Czerwinski, and Mary Czerwinski. Exploring 3D navigation: Combining speed-coupled flying with orbiting. In *Proc. CHI*, pages 418–425, New York, 2001. ACM.

[TSS+18] Matthias Thöny, Raimund Schnürer, René Sieber, Lorenz Hurni, and Renato Pajarola. Storytelling in interactive 3D geographic visualization systems. *ISPRS International Journal of Geo-Information*, 7(3):123:1–123:14, March 2018.

[Tuf01] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Second edition, 2001.

[TWL] David Turner, Robert Wilhelm, and Werner Lemberg. FreeType library. `https://freetype.org`.

[Var14] Johanna Varner. Scientific outreach: Toward effective public engagement with biological science. *BioScience*, 64(4):333–340, March 2014.

[vdZLBI11] Matthew van der Zwan, Wouter Lueks, Henk Bekker, and Tobias Isenberg. Illustrative molecular visualization with continuous abstraction. *Computer Graphics Forum*, 30(3):683–690, 2011.

[VFSG06] Ivan Viola, Miquel Feixas, Mateu Sbert, and M. Eduard Gröller. Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):933–940, September/October 2006.

[VFSH01] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference 2001*, VMV '01, page 273–280. Aka GmbH, 2001.

[VG05] Ivan Viola and M. Eduard Gröller. Smart visibility in visualization. In *Proc. CAe*, pages 209–216, Goslar, Germany, 2005. Eurographics Assoc.

[VGHN08] Pere-Pau Vázquez, Timo Götzelmann, Knut Hartmann, and Andreas Nürnberger. An interactive 3D framework for anatomical education. *International Journal of Computer Assisted Radiology and Surgery*, 3(6):511–524, August 2008.

[VI18] Ivan Viola and Tobias Isenberg. Pondering the concept of abstraction in (illustrative) visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(9):2573–2588, 2018.

[VKG04] Ivan Viola, Armin Kanitsar, and M. Eduard Gröller. Importance-driven volume rendering. In *Proc. Visualization*, pages 139–146, Los Alamitos, 2004. IEEE Computer Society.

[VWN03] Jarke J. Van Wijk and Wim A. A. Nuij. Smooth and efficient zooming and panning. In *Proc. InfoVis*, pages 15–22, Los Alamitos, 2003. IEEE Computer Society.

[WF97] Colin Ware and Daniel Fleet. Context sensitive flying interface. In *Proc. I3D*, pages 127–130, New York, 1997. ACM.

[WH07] Michael Wohlfart and Helwig Hauser. Story telling for presentation in volume visualization. In *Proc. VisSym*, pages 91–98, Goslar, Germany, 2007. Eurographics Assoc.

[WHA07] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, November/December 2007.

[WL96] Catherine M. Wilson and Suresh K. Lodha. Listen: A data sonification toolkit. In *Proc. ICAD*, Atlanta, 1996. Georgia Institute of Technology.

[WLW+16] Nicholas Waldin, Mathieu Le Muzic, Manuela Waldner, M. Eduard Gröller, David Goodsell, Ludovic Autin, and Ivan Viola. Chameleon dynamic color mapping for multi-scale structural biology models. In *Eurographics Workshop on Visual Computing for Biology and Medicine*, 2016.

[Wol09] Alexander Wolff. The map-labeling bibliography, 2009. http://i11www.iti.uni-karlsruhe.de/map-labeling/bibliography/.

[WTPA17a] Hsiang-Yun Wu, Shigeo Takahashi, Sheung-Hung Poon, and Masatoshi Arikawa. Introducing leader lines into scale-aware consistent labeling. In *Advances in Cartography and GIScience*, pages 117–130, 2017.

[WTPA17b] Hsiang-Yun Wu, Shigeo Takahashi, Sheung-Hung Poon, and Masatoshi Arikawa. Scale-adaptive placement of hierarchical map labels. In *EuroVis 2017 - Short Papers*, 2017.

[YLT18] Anders Ynnerman, Jonas Löwgren, and Lena Tibell. Exploranation: A new science communication paradigm. *IEEE Computer Graphics and Applications*, 38(3):13–20, 2018.

[Zha09] Xiaolong Zhang. Multiscale traveling: Crossing the boundary between space and scale. *Virtual Reality*, 13(2):101–115, June 2009.

[ZPLC15] Xiao Zhang, Sheung-Hung Poon, Minming Li, and Victor C.S.Lee. On maxmin active range problem forweighted consistent dynamic map labeling. In *Proceedings of the 7th International Conference on Advanced Geographic Information Systems, Applications, and Services*, pages 32–37, 2015.

Curriculum Vitae

David Kouřil

Nationality: Czech	david.kouril@gmail.com
Date of birth: November 26, 1991	+420 730 902 983
Place of birth: Zlín (Czech Republic)	www.davidkouril.com

SUMMARY / PROFILE

David Kouřil is a researcher with strong technical skills and a formal background in computer science. He specializes in applications of real-time graphics in scientific visualization where he's interested in interactions with complex data and virtual environments. In his graduate studies, he focused on biological data, developing techniques that make this data more accessible and understandable to general audience.

EDUCATION

Mar 2017 - (ongoing): **Doctoral Degree** (Dr.techn.) @ TU Wien (AT) 🇦🇹

- Topic: Interactive visualization of multiscale biological data
- Supervisor: Ivan Viola
- Highlight: IEEE Vis 2018 Best Paper Honorable Mention (SciVis)

Sep 2014 - Feb 2017: **Master's Degree** (Mgr.) @ Masaryk University (CZ) 🇨🇿

- Field: Computer Graphics
- Thesis: "Maya2CellVIEW: Integrated Tool for Creating Large and Complex Molecular Scenes" (graded A / Excellent)

Sep 2011 - Jun 2014: **Bachelor's Degree** (Bc.) @ Masaryk University (CZ) 🇨🇿

- Field: Computer Graphics and Image Processing
- Thesis: "Fast region labeling of binary images" (graded A / Excellent)

EXPERIENCE

Mar 2017 - now: **Project Assistant** @ TU Wien

- Main researcher for 3 research projects
  - Labels on Levels: labeling for molecular scenes
  - HyperLabels: multi-scale navigation
  - Molecumentary: virtual tours of molecular models
- Provided support to students and external collaborators with the Marion library
- Collaborated with international experts from both visualization and biology domain

Jul 2016 - Feb 2017: **Project Assistant Without Degree** @ TU Wien

- Integrated several research prototypes into a unified demo which was then submitted to the VIZZIES challenge (organized by National Science Foundation)
- Ported a high-performance molecular rendering technique (cellVIEW) from DirectX to OpenGL for a new proprietary library (called Marion)
    - This code was later used in commercialization of the Marion library by a spin-off company Nanographics GmbH
- Implemented nano-scale rendering of microtubules for project contracted by Allen Institute For Cell Science

Jun 2015 - Sep 2015: **QA Intern** @ Solarwinds

- Summer internship, performed both manual and automated software testing

Feb 2015 - Jun 2015: **Seminar Lecturer** @ Masaryk University

- Delivered seminar lectures in the course PV112 Computer Graphics APIs

Aug 2012 - Apr 2013: **C++ programmer** @ Celebrio

- Participated in development of a mobile game for tablet devices

TALKS

HyperLabels: Browsing of Dense and Hierarchical Molecular 3D Models (conference paper presentation) @ IEEE Vis 2020, Salt Lake City (USA), October 2020 (given remotely). [recording]

Navigating and Exploring 3D Biological Environments (invited talk) @ Visualization II course, Masaryk University, Brno (CZ), April 2020 (given remotely).

Navigating and Exploring 3D Biological Environments @ CellVis Summit, KAUST (Saudi Arabia), November 2019. [recording]

Labels on Levels: Labeling of Multi-Scale Multi-Instance and Crowded 3D Biological Environments (conference paper presentation) @ IEEE Vis 2018, Berlin (DE), October 2018. [recording]

Challenges and advances in multi-scale biology data visualization (invited talk), Czech Technical University, Prague (CZ), November 2017.

COMMUNITY SERVICE

Reviewing: IEEE Vis 2018 (InfoVis), EuroVis 2019, CESCG 2019, IEEE TVCG (Jan2020), IEEE Vis 2020 (InfoVis, SciVis)

EuroVis 2018: Fast Forward Chair, Student Volunteer

PUBLICATIONS

D. Kouřil, O. Strnad, P. Mindek, S. Halladjian, T. Isenberg, M. E. Gröller, I. Viola, “Molecumentary: Scalable Narrated Documentaries Using Molecular Visualization”, November 2020, preprint, arXiv:2011.02418 [cs.HC].

D. Kouřil, T. Isenberg, B. Kozlíková, M. Meyer, M. E. Gröller, I. Viola, “HyperLabels: Browsing of Dense and Hierarchical Molecular 3D Models”, *IEEE Transactions on Visualization and Computer Graphics*, Early Access, accepted February 2020. Doi: 10.1109/TVCG.2020.2975583

S. Halladjian, H. Miao, **D. Kouřil**, M. E. Gröller, I. Viola, T. Isenberg, "ScaleTrotter: Illustrative Visual Travels Across Negative Scales", *IEEE Transactions on Visualization and Computer Graphics*, 26(1):654-664, January 2020. Doi: 10.1109/TVCG.2019.2934334

H. Miao, T. Klein, **D. Kouřil**, P. Mindek, K. Schatz, M. E. Gröller, B. Kozlíková, T. Isenberg, I. Viola, “Multiscale Molecular Visualization”, *Journal of Molecular Biology*, 431(6):1049-1070, March 2019. Doi: 10.1016/j.jmb.2018.09.004

D. Kouřil, L. Čmolík, B. Kozlíková, H-Y. Wu, G. Johnson, D. Goodsell, A. Olson, M. E. Gröller, I. Viola, “Labels on Levels: Labeling of Multi-Scale Multi-Instance and Crowded 3D Biological Environments”, *IEEE Transactions on Visualization and Computer Graphics*, 25(1):977-986, January 2019. Doi: 10.1109/TVCG.2018.2864491

T. Koch, **D. Kouřil**, T. Klein, P. Mindek, I. Viola, “Semantic Screen-Space Occlusion for Multiscale Molecular Visualization”, *Eurographics Workshop on Visual Computing for Biology and Medicine*, 197-201, September 2018. Doi: 10.2312/vcbm.20181245

P. Mindek, **D. Kouřil**, J. Sorger, D. Toloudis, B. Lyons, G. Johnson, M. E. Gröller, I. Viola, “Visualization Multi-Pipeline for Communicating Biology”, *IEEE Transactions on Visualization and Computer Graphics*, 24(1):883-892, January 2018. Doi: 10.1109/TVCG.2017.2744518