Molecumentary: Adaptable Narrated Documentaries Using Molecular Visualization

David Kouřil, Ondřej Strnad, Peter Mindek, Sarkis Halladjian, Tobias Isenberg, M. Eduard Gröller, Ivan Viola

Molecumentary: Adaptable Narrated Documentaries Using Molecular Visualization

David Kouřil, Ondřej Strnad, Peter Mindek, Sarkis Halladjian,
Tobias Isenberg, M. Eduard Gröller, Ivan Viola

Abstract—We present a method for producing documentary-style content using real-time scientific visualization. We introduce molecumentaries, i. e., molecular documentaries featuring structural models from molecular biology, created through adaptable methods instead of the rigid traditional production pipeline. Our work is motivated by the rapid evolution of scientific visualization and it potential in science dissemination. Without some form of explanation or guidance, however, novices and lay-persons often find it difficult to gain insights from the visualization itself. We integrate such knowledge using the verbal channel and provide it along an engaging visual presentation. To realize the synthesis of a molecumentary, we provide technical solutions along two major production steps: (1) preparing a story structure and (2) turning the story into a concrete narrative. In the first step, we compile information about the model from heterogeneous sources into a story graph. We combine local knowledge with external sources to complete the story graph and enrich the final result. In the second step, we synthesize a narrative, i. e., story elements presented in sequence, using the story graph. We then traverse the story graph and generate a virtual tour, using automated camera and visualization transitions. We turn texts written by domain experts into verbal representations using text-to-speech functionality and provide them as a commentary. Using the described framework, we synthesize fly-throughs with descriptions: automatic ones that mimic a manually authored documentary or semi-automatic ones which guide the documentary narrative solely through curated textual input.

Index Terms—Virtual tour, audio, biological data, storytelling, illustrative visualization.

1 Introduction

Scientific visualization helps researchers to make sense of their data. Visualization today also contributes to another, increasingly important part of science: scientific outreach [64]. A growing number of researchers now focus on communicating the current state-of-the-art of life sciences to students and stakeholders, and also to the general population. Many visualization techniques for biology 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.

Visual representations can only provide insights into scientific data if the viewer is familiar with the concepts of the particular field. A plethora of written materials exists in the life sciences (e. g., textbooks, online educational sites) with detailed information about the studied topic. In these media, 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 nowadays [14]. Scientific concepts are presented using computer-generated animations on sites, such as YouTube or Vimeo. These videos communicate a topic in an engaging way by leveraging storytelling techniques developed over decades by the animation industry. A verbal narration is an essential part of the educational content's explanatory value.

Yet, pre-rendered computer animations are significantly different from interactive 3D visualizations. 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 [6]. The existing cases of applying visualization in science communication underline the need for incorporating explanation and guidance for public dissemination.

Our work is motivated by large molecular models, e. g., of viruses and bacteria. 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. We pose the following research question: *How can explanatory information about the function and role of individual subparts be integrated into a 3D visualization?*

We address this question with a method to elevate 3D scientific visualization into a scientific documentary (Figure 1). The explanatory information is integrated through verbal annotation using the auditory channel. We couple

- *D. Kouřil is with Masaryk University, Czech Republic, and TU Wien, Austria. E-mail: dvdkouril@cg.tuwien.ac.at.*
- *O. Strnad and I. Viola are with King Abdullah University of Science and Technology (KAUST), Saudi Arabia.*
- *P. Mindek is with TU Wien and Nanographics GmbH, Austria. M. E. Gröller is with TU Wien and the VRVis Research Center, Austria.*
- *S. Halladjian and T. Isenberg are with Université Paris-Saclay, CNRS, Inria, LISN, France.*

Manuscript received Apr. 1, 2021; revised Nov. 8, 2021; accepted Nov. 15, 2021. Author version. DOI: 10.1109/TVCG.2021.3130670

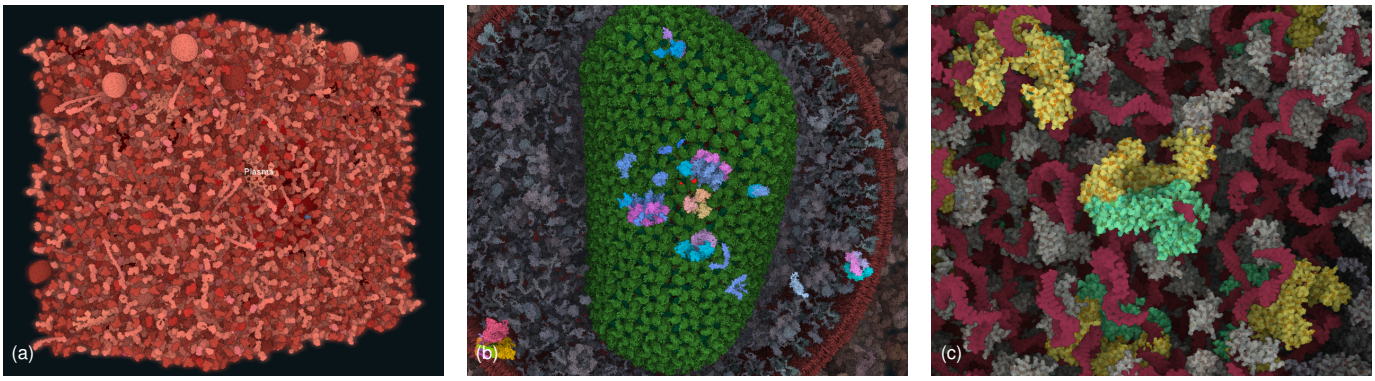

Fig. 1. In the tour of the HIV in blood plasma model (a) we, for example, visit the capsid (b) which contains the genetic information of the virus. Besides the RNA, the capsid contains several important proteins, such as Reverse Transcriptase (c).

verbal annotations (i. e., the commentary) with an automatic fly-through of a 3D structural model, providing visuals relevant to the commentary. The annotation communicates the roles and functions of the building blocks of the model (e. g., proteins), resulting in a virtual guided tour of the particular 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 (e. g., proteins assembled into protein complexes) and the functional relationships (e. g., molecular interactions) 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 adaptable 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 *adaptable documentary* framework that comprises real-time methods for producing scientific documentaries in an adaptable and future-proof way;
- *molecumentaries* as an exemplary application of the adaptable 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.

2 Related Work

Our work furthers efforts in utilizing data visualization for science dissemination. In doing so, we touch upon storytelling using visual data representations to communicate facts and stories embedded in the data. A generated voice-over is another integral part of our conceptual framework. We, therefore, review the utilization of audio in the visualization field. We also couple the verbal commentary with camera animation and dynamically resolving the occlusion of focused model parts. Camera control and occlusion management are thus two additional topics of our related literature review.

2.1 Visualization for Science Outreach

Visualization plays a vital role in disseminating scientific concepts to a broader audience. While interactive visualization tools offer participatory and highly engaging learning, they are usually designed for expert users, often with a high complexity and a steep learning curve. Alternatively, modern computer graphics provides tools for authoring computer animations, through which a skilled storyteller can communicate concepts at an appropriate knowledge level. Given the need of specific domain expertise to produce content that communicates scientific findings, a new job role has emerged—a science animator, as described by Iwasa [25]. Real-time graphics have recently reached a fidelity high enough to diminish the need for the lengthy and costly manual animation and rendering process. Ynnerman et al. [69] coin the portmanteau *exploranation* to describe the practice of using real-time visualization tools both in research (i. e., *exploratory* tasks) and scientific outreach (i. e., *explanatory* tasks). Their work has found successful applications, e. g., in astronomy dissemination [6]. Such applications can often be deployed as museum or science center exhibits [23], [40] and facilitate broad audience learning.

Visualization for education is not a new concept. Already in the 1990s, researchers have designed systems to depict anatomical models with textual annotations, to help students connect the learned material with visual depictions (e. g., Preim et al. [53]). These works continue in the long tradition of educating medical students through visual means. Works with traditional learning materials, e. g., the textbook *Gray's Anatomy*, have also been augmented by modern technologies [65]. Thanks to advances in real-time visualization, researchers can now show progressively larger and more complex 3D models, often utilizing visually pleasing animations [52]. With these advanced tools at their disposal, visualization designers can guide inexperienced users through a complex dataset [9] or tell an engaging story hidden in the data.

2.2 Visualization Storytelling

The role of storytelling in visualization remains an open research topic, with several researchers investing efforts to

define it. Kosara and Mackinlay [30] debate the potential of storytelling in visualization research. Lee et al. [34] discuss how storytelling should be scoped within the visualization context. Tong et al. [62] provide a recent survey of visualization literature employing storytelling elements.

Segel and Heer's landmark paper [57] analyzes existing cases of telling a story through data visualization. They define several "genres" of narrative visualization, but their work mostly focuses on *information visualization*. Hullman and Diakopoulos [24] further investigate rhetorical devices available for narrative visualization. Many other examples incorporating storytelling aspects in information visualization include Kwon et al. [32], Ren et al. [54], Gratzl et al. [21], and Gershon and Page [18].

Ma et al. [41] provide insight into how storytelling in the context of *scientific visualization* may be conducted. Even earlier, Wohlfart and Hauser [68] explore the continuous spectrum between fully interactive and fully story-told presentation of a volumetric model. A big part of storytelling in the scientific visualization context is the animation of both camera and visual mapping attributes. These animations can either use templates [1] or leverage an inventory of previous user interactions [36]. Storytelling techniques can be found in visualizing several science areas: e. g., geology [37], geography [60], medicine [68], and biology [59].

In the real-time data visualization environment, storytelling is highly related to *interactive storytelling* investigated mainly in the Virtual Reality (VR) community. Novel challenges arise if the user can look in any direction, making it harder to present a specific narrative. Researchers, e. g., Glassner [19] and Perlin [51], previously pointed out the differences between narratives in the traditional media, e. g., books and movies, and the new media, e. g., computer games.

2.3 Audio in Visualization

Storytelling has a tradition of verbal, and particularly oral, form. However, the use of audio in visualization storytelling has been limited. Munzner's visualization textbook [46] indicates reasons. With visual processing, a person can quickly gain an overview with one glance. This is not possible with the inherently linear processing of audio. The dual coding theory [12] suggests that humans process visual and verbal signals via separate systems. This implies that a combination of visual and auditory inputs can lead to augmented cognition. In our work, we leverage this finding and provide both visual and audio aspects for describing biological structures.

In principle, there are two ways of including sound in data visualization: *sonification* and *voice-overs*. Data sonification refers to the process of mapping data attributes to non-speech audio. Several sonification toolkits have been developed, such as Porsonify [42] or Listen [67], but this practice remains a niche in the data visualization community. Voice-overs, traditionally pre-recorded by a voice artist, contain human speech and most often provide commentary auxiliary to a visual presentation. In recent years, methods for generating artificial speech from a textual input (text-to-speech) have significantly improved [29], [58]. We leverage these advances in our work and generate synthetic voice-over using a text-to-speech library.

In scientific documentaries, the audio commentary is matched with changing visuals. There are primarily two components in three-dimensional scenes that can help to reveal or highlight specific objects: camera control and occlusion management as we discuss next.

2.4 Camera Control

The position and orientation of a virtual camera greatly influence the perception of a 3D scene. Much effort has been devoted to the problem of camera control in various virtual environments. Christie et al. [11] provide an overview of navigational methods. Several authors discuss the challenges for a camera navigating a *multi-scale* environment [43], [70], which also applies in our case. In contrast, methods for intuitive navigation in a 2D environment, e. g., van Wijk and Nuij [63], often cannot be easily transferred to the 3D case.

Several researchers utilize *cinematographic* concepts and rules to model visually pleasing camera movements. Burtnyk et al.'s StyleCam [7] and ShowMotion [8] works provide examples of such techniques. Lino et al. [38] further expand the use of cinematography on scenes featuring actors and interactions between them. Such scenes, where characters and human-scale environments drive the narrative, are distinctly different from narratives based on data visualizations. Amini et al. [2] analyze *data videos*, i. e., motion graphics animation utilizing—mostly 2D—data visualizations. Their approach is similar to Segel and Heer's [57] work on narrative visualizations. Virtual cameras allow techniques impossible in reality, such as spawning very many cameras and compiling their viewpoints into a summary, e. g., of a multiplayer computer game [45].

In principle, the more a camera follows cinematographic principles, incorporates guidance [9] or constraints [22], or captures a specific narrative, the more it transfers control from the user to an *automated* approach. Automated camera control has been explored since the beginning of computer graphics, with Jim Blinn's work [5] as an early example. Christie et al. [10] provide a general overview for automated camera planning. Research dealing with automatic camera control overlaps with techniques for robot or drone navigation [17], [48]. Salomon et al. [56] build on research in robotics and present an approach for planning a collision-free path between two points in a complex 3D model, using a global roadmap and a reachability analysis. Oskam et al. [50] further include visibility conditions into the path planning.

While the techniques for navigating in a macroscopic world can serve as an inspiration, their applicability is limited in virtual environments composed of spatial scientific data, which often represent structures on the nano- to micro-scale. An example of the added complexity is the usual density of 3D scientific data, e. g., volumetric models from CT or molecular models, which leads to frequent occlusions.

2.5 Occlusion Management

In principle, occlusion is eliminated by changing either spatial or optical attributes of scene objects [15]. We consider the spatial arrangement of objects to be important for molecular models. Thus, we reduce occlusion by changing the visual features, i. e., using visibility settings to remove objects occluding those in focus. Cut-aways are frequently

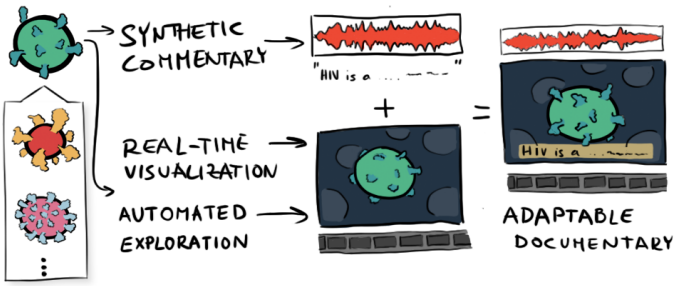

Fig. 2. The *adaptable documentary* concept: We provide visuals as a real-time visualization and couple it with an automated exploration of the 3D scene. We augment the fly-through with a synthetic commentary that we generate on-demand, as opposed to using a pre-recorded voice-over.

used to reveal occluded parts of a 3D model. Li et al. [35] present a sophisticated method to semi-automatically resolve cut-away views for complex polygonal models. Occlusion-handling techniques often leverage domain or specific model characteristics in visualization applications [66]. For large molecular models the *multi-instance* aspect is used. There are multiple copies (i. e., instances) of each structural type. Removing some of the instances leads to a reduced density and, therefore, also occlusion. Le Muzic et al.'s visibility equalizer [33] employs cutting objects and allows the user to adjust how molecular instances are cut away.

The occlusion management method in this paper is based on our previous work [31], where we presented a method for *sparsifying* a dense molecular scene. We employ a cutting plane where some scene objects are exempt from being cut away. This approach has in the past been also called selective clipping [39] or selective cutting [61]. Birkeland et al. [4] 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.

3 Adaptable Documentary: Overview

To address needs in science communication, we propose *adaptable documentaries* (Figure 2), i. e., a conceptual framework in which we use real-time visualization as a medium for 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 emphasize adaptability, i. e., the ability to adjust to future inputs. Our framework rests on three components: the use of *real-time visualization* instead of pre-rendered animations, *automated exploration* to procedurally traverse and showcase 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 eliminate the lengthy rendering process. 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.

Automated Exploration: A dynamically generated presentation of an arbitrary 3D model cannot rely on pre-authored camera animations. In adaptable documentaries, we instead use methods for automated exploration to showcase the model. Automated and guided exploration is complex, and researchers recognize the enormous number of options—between giving full control to users and completely stripping them of any control. For our initial version of adaptable documentaries, we focus on the variant where the exploration is fully automated. Automating the exploration has several other advantages: an algorithmic approach can be tailored to not miss salient structures—which a novice user may—or it can incorporate storytelling elements to present a more coherent storyline.

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 adaptable documentaries. Flexibility is needed in this context, as new knowledge is likely to be discovered and will need to be incorporated in the future. We use a procedural approach to provide a 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 an adaptable 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 the adaptable documentary concept to be applicable in several scientific domains. We described the components on a high level, allowing us and others to tailor the specific implementation to a particular domain, e. g., volumetric medical datasets. For the remainder of this article, however, we demonstrate it in the context of large molecular models. As a proof-of-concept we produce an adaptable documentary movie that integrates additional domain knowledge and provides explanation to novices.

We use specific molecular models that are assembled using mesoscale modeling [26], [27]. This process defines hierarchical compartments, their building components (i. e., molecules), and concentrations of these based on past scientific observations. A packing algorithm then fills the compartments with molecular instances. The resulting model serves as the input to our method and contains, for each molecular instance, position, orientation, and type. For the latter, we refer to a Protein Data Bank entry via a PDBID, i. e., a 4-character alphanumeric molecule identifier. Furthermore, we extract the model's hierarchical organization from its decomposition into compartments.

Before we explain the technical details of how we generate a molecumentary from these input models, 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, basing our terminology on that of story-telling theoretician Olaf Bryan Wielk (`https://www.beemgee.com/blog/story-vs-narrative/`). 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 story and narrative, as shown in Figure 3.

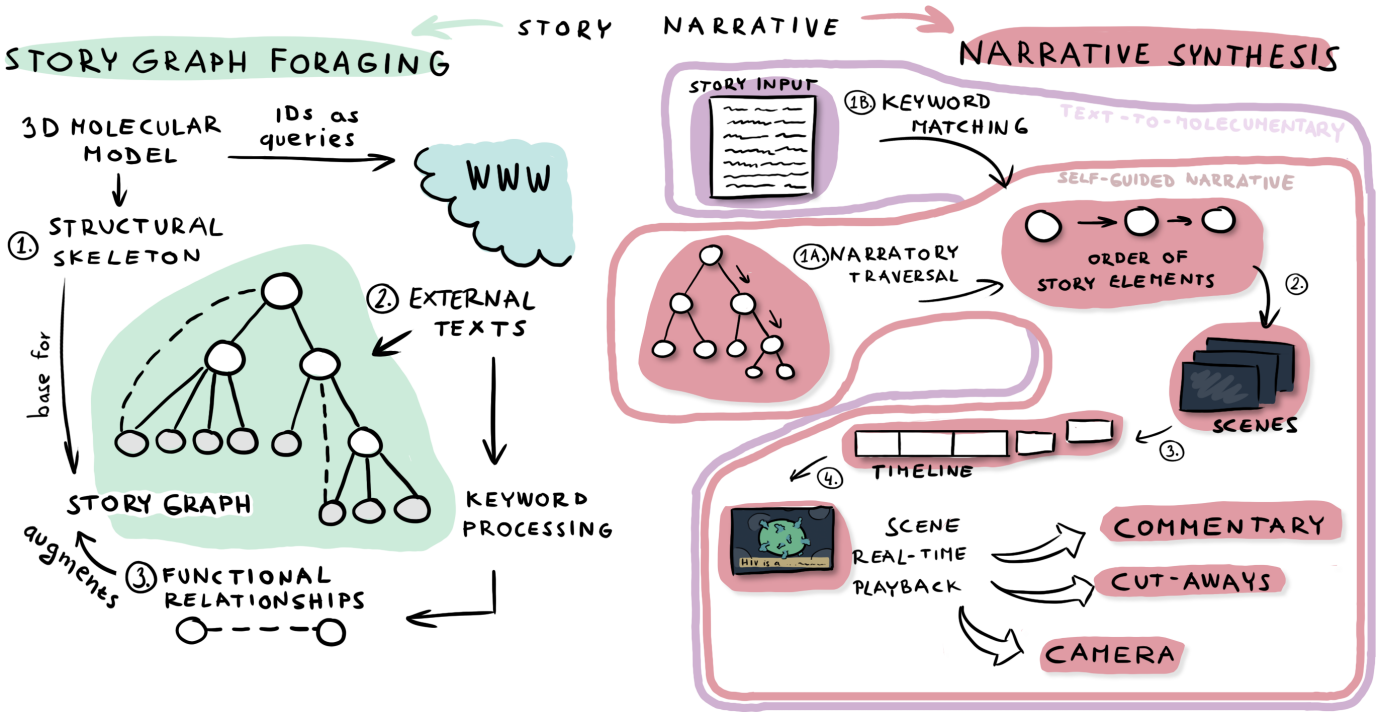

Fig. 3. Overview of the framework for generating *molecumentaries*. In describing the technical contributions we follow the distinction between a story (static representation of story elements) and a narrative (dynamic arrangement of these story elements). *Story graph foraging* provides a scalable approach for compiling information about the model that can be used for storytelling, while the real-time *narrative synthesis* encompasses solutions for turning the story graph into a specific narrative at runtime.

In Section 4, we explain how we organize a story in a data structure called *story graph*. Story graphs are often utilized in interactive storytelling [55]. In our case, it holds all the model elements, their relationships, and verbal descriptions of the biological model parts. Creating such a story structure manually is tedious and, in the context of a molecumentary, would require the involvement of a domain expert. We thus 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 external sources, and then extract relations between the components from these textual descriptions.

In Section 5, we then show how we generate the actual molecumentary. We use the story graph to produce 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 shown model elements in two ways. In the first case, the molecumentary 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 sees its impact.

Our concept facilitates adaptable science communication. By automatizing a large portion of the scientific movie production pipeline, we are able to immediately incorporate new knowledge, e. g., new research results, into science communication such as scientific movies, interactive learning tools, or museum installations. While we do deal with stories and narratives, we do not attempt to provide a solution to the problem of generative storytelling. We rely on texts coming from various writers, but essentially consider these texts as "black boxes." We do not extract meaning and do not aim to produce creative stories that are stylistically correct, even though this could be a useful future extension.

4 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 into a multigraph. The edges represent relationships between the structural elements. These relationships can be of several types as well. 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 otherwise 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

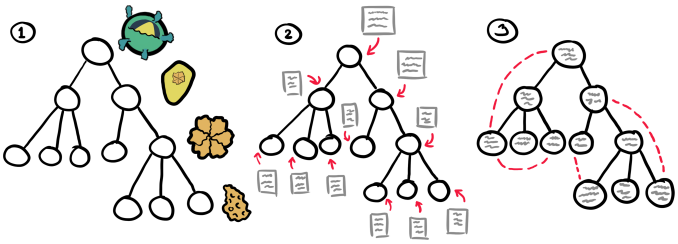

Fig. 4. The three steps in story graph foraging. First, we build the structural skeleton. Second, we associate textual descriptions with individual nodes. Finally, we add functional relationships.

foraging rather than construction to express the flexibility and liveliness of this process. The story graph is not only constructed once with a single 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 4), each improving the possible generated narrative.

4.1 Step 1: Structural Skeleton Foraging

In the first step, we build the basic structural skeleton of the story graph from the hierarchical organization of the input model. We mirror each molecular type in the model as a leaf node in the story graph. We add further nodes for higher-level composite objects based on the hierarchical compartments of the input model. For example in the HIV model (indicated in Figure 4), the capsid envelope is formed from capsid proteins, which first assemble into polymers. The polymers in turn build the capsid. The capsid protein is therefore added as a leaf node and the hierarchical assembly is modeled through inner nodes in the story graph.

At this point, the names of individual components are the only descriptive information that we can relay to the viewer, and show as textual labels. Furthermore, a simple narrative can be synthesized using even a story graph just containing the structural skeleton. However, the output would be rather rudimentary, communicating the structural organization of the biological model (e. g., "Structure X contains components Y, Z, and W. Let us look at Y first.").

4.2 Step 2: Type Node Descriptions Foraging

We can improve the initial rudimentary narrative by incorporating descriptions about the individual structure types, which explain the role of the associated structure in the biological model. The second step of story graph foraging is thus to gather these descriptions.

There are several options for getting the descriptions. First, some descriptive texts can be manually written and **supplied locally** along with the structural model. We use 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 use an alternative way of gathering descriptions. We take the standardized

"Capsid protein forms a cone-shaped coat around the viral RNA, delivering it into the cell during infection."

Fig. 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.

names of biological structures (e. g., "Albumin", ID: 1YSX) as keywords for searching in **external, online repositories**.

Publicly accessible databases 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 to fetch the structural description as a response. We target short, high-level descriptions that explain the searched term in a few sentences. This process can be done on demand and we do not need to pre-fetch the descriptions for the whole model before the narration starts, saving memory. One of the major benefits of real-time foraging is the scalability to models of arbitrary size, provided that they are reasonably annotated. Also, by fetching the data online in multi-lingual databases, we can query information in several languages. 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 an expressive paragraph of text. A label can also contain a name which may lead to an empty query response. In this case we fall back to the structural commentary, which we explain later in Section 5.1.3.

If 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.

4.3 Step 3: Functional Relationship Edges Foraging

So far the order of explanation can only be driven by the structural relationships, i. e., traversing the hierarchical organization. We want to relate structures that are associated not because of their proximity in the hierarchy, but rather because how they interact. To enable exploration along functionally related objects, we add functional relationships to the story graph—the final step of story graph foraging.

We could establish functional links by using data about the metabolic exchanges between structural components of the modeled organism, i. e., the metabolic pathways [28]. However, integrating such data would likely require a manual intervention from a domain expert.

We instead use another—text-based—approach to extract functional relationships, as illustrated in Figure 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 or downloaded texts from Section 4.2, split them into sentences, and search for keywords they may contain. If we detect keywords in a sentence, we establish functional relationship edges between structures associated with these keywords. We consider these functional neighbors when the story graph traversal selects the nodes to

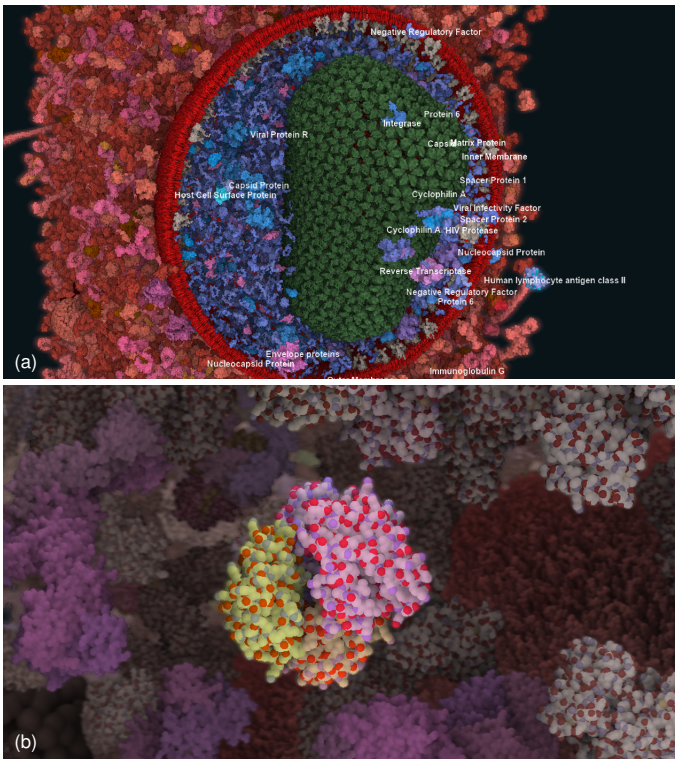

Fig. 6. Overview scene (a) communicates the composition of an object while a focus scene (b) describes its function. A transition scene is defined to switch focus and connect overview and focus scenes in the narrative.

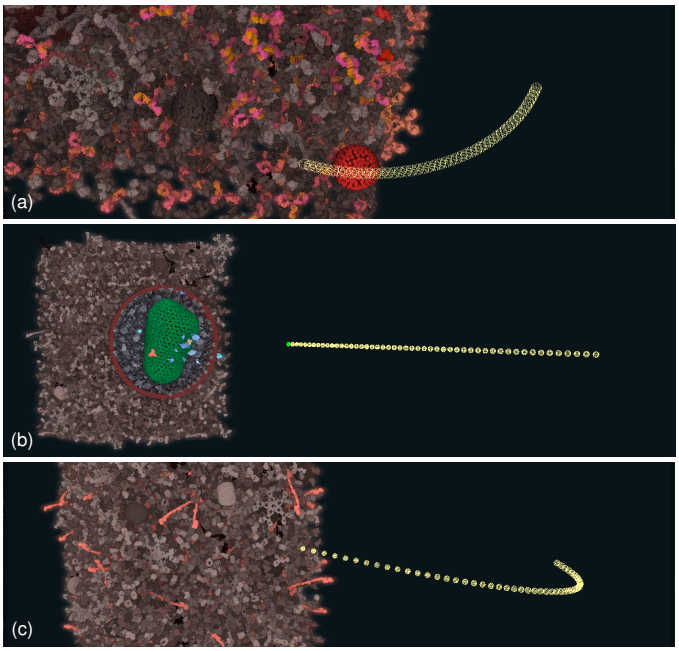

Fig. 7. Illustration of the three camera animation types used in a molecumentary synthesis: anchored orbiting (a), direct flying (b), and curved transition (c).

be covered in the synthesized narrative. The described initial approach for foraging functional relationships could, in the future, be improved by employing ontology-based methods.

5 Narrative Synthesis

After we created the story graph with both structural and functional information, we prepare the story for the narrative synthesis. 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 a human-authored, textual narrative and employ our molecumentary synthesis to produce accompanying visuals.

5.1 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 aspects of individual parts of the molecumentary. We use the timeline as a queue—we add (push) scenes to the back and remove (pop) them from the front, implementing a first-in-first-out approach.

We use three types of scenes: focus, overview, and transition. A *focus scene* (Figure 6(b)) 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 we use subtle rotation animations to provide parallax, and give a detailed description of the function and role of the focused object inside the modeled organism. A focus scene typically lasts as long as it takes for the speech synthesis engine to read out the descriptional text.

An 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. Hence, we incorporate overviews as a second scene type. An *overview scene* (Figure 6(a)) shows all building blocks of a certain model part to communicate the object's structural composition. We realize this by adjusting the cut-away settings of the view. We highlight representative instances for each subcomponent and place the camera to show all of them. For this purpose we use Kouřil et al.'s [31] bounding sphere approach. In the accompanying commentary we describe the components and explain their relationships with the current focus object.

To be able to meaningfully switch between the various focus and overview scenes, we also need animated transitions to communicate a shift of emphasis. *Transition scenes* connect overview and focus scenes and provide context for a fly-through. They usually contain significant camera movements and changes in the visual representation. In the verbal commentary of transition scenes we provide additional guidance and explain the view changes.

Next we describe the three processes—camera animation, occlusion management, and voice-over—that we used in implementing the scenes in the molecumentary synthesis.

5.1.1 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 7. Many more camera movements can be incorporated and developed for future

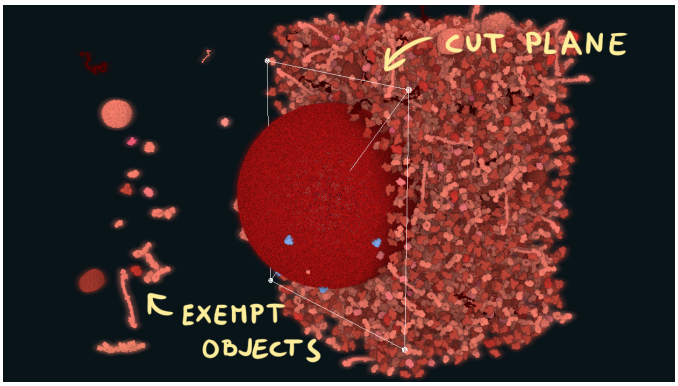

Fig. 8. Traveling cutting plane: We remove all objects—except a selected subset—that lie between the cutting plane and the camera position to reveal inside components of the model.

applications. Here, we describe our basic camera language sufficient to be used in our prototypical implementation.

To generate the camera animations, we start with the position and geometry of each scene's focus structures. We then approximate the target object's shape and size with a bounding sphere, which we can compute in real-time. In our molecular model scenario, spheres approximate the shape sufficiently. We create camera animations between targets, which we specify with two attributes each: world position and 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. We select the orbiting direction (clockwise or counterclockwise) randomly in each scene.

For a continuous narrative, 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 (start and target) are visible from the initial camera viewpoint. If the target position is outside the view frustum, direct flying can be suboptimal in communicating the spatial relation between the two objects.

Therefore, we introduce *curved path animation* as third movement type. 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 quadratic Bézier curve, but other curve types can be used as well.

We apply easing functions to the camera transitions for a smoother impression and visually more pleasing movement.

5.1.2 Occlusion Management

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

We employ a *traveling cutting plane* (see Figure 8). We define a cutting plane in the scene and do not render objects between the cutting plane and the camera. We exclude, however, 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 numbers of these objects in the model. The cutting plane *travels*, i. e., we animate it and the set of objects we show throughout the molecumentary. This successively reveals objects that are being verbally described in sequence. We perform these 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 frequency in the model. We then re-position the cutting plane to the center of a selected representative instance. The instance closest to the camera is selected as the representative, and we orient the cutting plane to be parallel to the viewing plane at the moment the object comes into focus, i. e., we orient it according to the camera's initial back vector.

An *overview scene* communicates the inner composition of a structure. Thus, a *transition scene* leading to an overview scene features an animation that opens up the structure of interest and reveals its inside. We do so by fetching the structural components (child nodes) of the focus structure and, for each of the child nodes, pick a representative instance and exempt it from the cutting. We place the cutting plane at the position of the representative furthest away from the camera so 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. This could make it less apparent that our scene changes communicate an opening of the model, as opposed to a change of the scene altogether. We also only use a single cutting plane in our design to avoid the complexity of managing multiple planes or even a plane hierarchy: It would be difficult to ensure that an object, selected later in the molecumentary, is not cut away.

5.1.3 Verbal Commentary

We realize the verbal commentary using text-to-speech synthesis. We assemble three types of commentary—structural, descriptional, and navigational—in textual form first and then turn them into speech using an artificial voice.

We use *structural commentary* in overview scenes to describe the structural composition of certain composite objects. An example of a structural commentary is *"Blood plasma consists of Hemoglobin and Heparin and others."* We construct the commentary procedurally based on the hierarchical object composition using sentence templates. We define basic sentence templates in an external file, which can be further extended. The basic templates to communicate hierarchical organization use phrases such as "consists of" or "belongs to", in combination with several pre-defined variables. We replace these variables in real-time with respective values based on the current story graph traversal. The variable *$name* denotes the element on which the story currently

focuses. Variables *$siblings*, *$children*, *$parent* contain hierarchical information related to the current node *$name*. Put together, an example of a template sentence is *"$name consists of $children"*. In large hierarchies, *$children* and *$siblings* can contain tens or hundreds of nodes—too many to list them all in the commentary. Therefore, we randomly select a subset (we use three) to keep the sentence short. Since we generate the commentary on-demand, in case the virtual tour returns back to the same node the structural commentary will be slightly different each time, providing a level of variety.

Next, 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 4.2) of the story graph nodes to synthesize the text to describe an object's functions and significance in the model. We use pre-defined texts with a higher priority than ones fetched from online sources. We currently consider the texts as black boxes, so their expressiveness depends on the authors and we use them as is. In the future, we envision that the texts could be further optimized for a better speech expression, e. g., by removing non-verbal signs that cause issues in the text-to-speech synthesis.

Finally, we use a *navigational commentary* in transition scenes. Its purpose is to contextualize what happens in the transition scene and to connect the overall narration. We synthesize the sentences using the same templating approach as in the structural commentary, but with a different set of templates. We introduce another template variable, *$previous*, which points to the node which has been in focus just before *$name*. An example of a transitional commentary template is: *"After focusing on $previous we can see $name."*

We also display textual labels in the scene [31] to connect the verbal narration with the shown structures and to help viewers to differentiate the mentioned objects. Dynamically placed labels name the structures of those representative instances that are relevant to the current scene.

5.2 Self-Guided Narrative

Given the general molecumentary synthesis, we now present two variants of this application of adaptable documentaries. First, we showcase a self-guided molecumentary, i. e., one in which we do not use input that would inform the narrative to be shown. Instead, we automatically create the fly-through based on the organization of the model and a specific *narratory traversal* story graph exploration.

5.2.1 Narratory Traversal

In deciding the order of story nodes in the documentary, we traverse the story graph that represents the hierarchical organization of the model. We wish to communicate the model organization to the viewer. In the context of the molecumentary synthesis we aim to replicate the look and feeling of a scientific movie. The traditional algorithms for traversing a tree or a graph data structure (e. g., depth-first or breadth-first search), however, do not provide the engaging results we desire and would, instead, result in a mechanical and rigid exploration.

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. We employ a stochastic approach, but other ways of transforming the story graph into narratives can be considered, e. g., Fujiwara et al.'s tree reduction method to replay the history of interaction in visual analysis [16]. 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 options pool contains the possibilities 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. These are (a) parent, (b) children, or (c) functionally related nodes. The nodes represent potential next targets, and we recompute the pool any time a node is pushed to or popped from the stack.

Algorithm 1: Next story node selection

```
// options from the pool
var options;
// times of last visit
var visitedTimes;
min ← getMinimumValue(visitedTimes);
foreach option ∈ options do
    if visitedTimes[option] = min then
        candidates.add(option);
    end
end
foreach c ∈ candidates do
    priority ← Priority(c);
    priorityRange ← priorityRange + priority;
end
rand ← random(0, priorityRange);
prioSum ← 0;
foreach c ∈ candidates do
    priority ← Priority(c);
    valA ← prioSum; valB ← prioSum + priority;
    prioSum ← prioSum + priority;
    if valA < rand ≤ valB then
        next ← c;
        break;
    end
end
visitedTimes[next] ← currentTime;
return next;
```

We stochastically pick the next targets from the pool, as detailed in Algorithm 1. Our selection criterion is whether the potential next node has been previously shown, and if so when. We specifically use the time of the last visit to ensure that we continuously traverse the whole model if the molecumentary is left to run for longer periods. We use a $Priority$ function to model a priority distribution among the nodes, and we define it as

$$Priority(n) = \begin{cases} P_{lower} & n \text{ is a leaf node} \\ P_{higher} & n \text{ is an inner node} \end{cases} . \quad (1)$$

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

5.2.2 Timeline Building and Playback

The step-wise procedure for determining which nodes will be featured in the narrative is not yet sufficient. To produce a molecumentary we still need to turn the node sequence

Algorithm 2: Scene generation (self-guided narrative)

```
lastScene ← timeline.last;
if lastScene.type = overview then
    transitionOverviewToFocus(current, next);
else
    transitionSiblings(current, next);
end
focus(next);
if isLeaf(next) = false then
    pushToStack(next);
    transitionFocusToOverview(next);
    pushOverview(next);
end
```

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 first add a transition scene from the current object in focus to the new one onto 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., 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 scenes to introduce the coming composition explanation. We detail the procedure in Algorithm 2.

5.3 Text-To-Molecumentary

Often there already exists a human-authored description of a particular model that describes the important parts and their functional behavior. Our second synthesis variant uses a story in a text form as an 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 there is a match. To prevent frequently mentioned keywords 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 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 done in the self-guided version. We then push these scenes to the timeline, which we later play in the same manner as explained before. 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 exhibit 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. 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 3.

Algorithm 3: Scene generation (text-to-molecumentary)

```
// list of sentences from the text
var sentences;
// set of previously used keywords
var usedKeywords;
foreach s ∈ sentences do
    keywords ← identifyKeywords(s);
    keyword ←
      selectFirstNotIn(usedKeywords, keywords);
    current ← getType(keyword);
    if isLastSentence(s) then
        child ← current.root.children[0];
        transitionOverviewToFocus(child);
        transitionFocusToOverview(child);
        overviewScene(child);
    else if hasChildren(current) then
        transitionFocusToOverview(current);
        overviewScene(current);
    else
        if previous.parent ≠ current.parent then
            transitionSiblings(current.parent, current);
            focusScene(current);
        else
            transitionSiblings(current, current);
            focusScene(current);
        end
    end
    usedKeywords.insert(keyword);
    previous ← current;
end
```

6 Results

We developed a prototypical implementation of the molecumentary synthesis on top of the Marion library [44], which supports biology communication. The molecular rendering uses cellVIEW's [47] 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. In our exemplary implementation, we use Wikipedia's API to fetch short descriptions of the keywords, called *extracts*. The response time for such a query was ≈150 ms—well within the limits of a live production. In the majority of our tests, three sentences from the extracts are sufficient to describe a structure. The result 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.

The framework's component for verbalizing texts is implementation-agnostic. Marion is based on Qt, so we leverage its text-to-speech functionality. Qt's Speech component [13] 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, i. e., Google's Cloud Text-to-Speech API [20]. It allows us to customize several speech attributes and, in our experience, produces more natural sounding output than the OS libraries. To support languages other than English, we take a user-defined keyword translation. These translated keywords allow us to retrieve relevant information in the target language.

We produced exemplary molecumentaries—which we provide as supplementary videos—from three molecular datasets. We recorded all the videos in real-time at FullHD resolution with a performance of approx. 15 FPS (in zoomed-

in scenes) to 30 FPS (overview scenes).

First, we use a **HIV in blood plasma** dataset (Figure 1) from Scripps Research. The structural model contains ≈18,500 protein instances, 200,000 lipids, and a single RNA strand. Its story graph consists of 53 nodes—45 protein types, two lipid types, a single RNA type, and five higher-level nodes. 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. 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 (Video A) is 2:42 minutes long. We also produced a self-guided movie (Video B), which we stopped after 4:33 minutes. In this time, our framework visited 11 story graph nodes.

Second, the **Mycoplasma dataset** (Figure 9) has also been provided by Scripps Research. The relatively smaller structural model comprises ≈5,400 protein instances. The story graph contains 22 nodes overall—15 protein types, four strand types, and three higher-level nodes. Because it is a preliminary model it does not yet contain a lipid membrane. Approximately half of the proteins are well annotated and the model contains no additional textual descriptions. For this reason, we fetched all needed descriptive texts from Wikipedia and produced a self-guided movie (Video C). The sample movie visits 11 nodes in 4:35 minutes.

Finally, we used the **SARS-CoV-2** dataset (Figure 10) provided by KAUST [49]. It consists of ≈3,200 protein instances, ≈180,000 lipids, and an RNA strand. This model's story graph contains 18 nodes—five protein types, four lipid types, one RNA strand represented by its five building blocks, and four higher-level nodes. The model is annotated with human readable labels, but no predefined textual description is available. We retrieved all textual descriptions from Wikipedia and only produced a self-guided movie (Video D). It is 3:20 minutes long, visiting 10 story graph nodes. 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 ends up zooming in too much. The resulting view is not very attractive, and we need to address this in the future.

The sample molecumentary recordings provided as supplementary material demonstrate raw results of the described framework. As such, they contain suboptimal moments that could be corrected or improved by human intervention in the traditional pipeline. Our motivation was to define automatic methods that work well enough in the majority of cases. Therefore, we provide the videos as outputted by our method and leave improvements for future work.

7 Discussion

To reflect on our work and validate its utility in biology communication, we showed the results to two domain experts as well as to a high school teacher in biology.

The domain experts—with 45 resp. 33 years of professional experience—appreciated that our approach is able to showcase complex molecular models. They confirmed that the many parts of the 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. According to them, our method would make a valuable tool for semi-automated content creation, provided that we add more user interaction in the creation pipeline.

In a second interview, we talked to a high school teacher in biology who currently works in secondary education of grades 9 to 12. She holds a PhD in virology and had worked in academic research for five years, before starting to teach science. She noted that, besides the usual teaching tools, such as textbooks, frontal lectures, and manipulatives, she often employs scientific animations from WEHI (`https://www.wehi.edu.au/wehi-tv`). She noted that videos that employ 3D graphics can help to contextualize the diagrammatic representation from textbooks, which can give a false impression to young students. She also described two main issues with the usage of scientific animations in teaching. First, it is impossible to interrupt them and gain more detailed views. Second, if a video does not follow the syllabus, it has a lower utility.

While our framework currently does not implement it, it would be easy to add the ability to interrupt a molecumentary and allow viewers to directly manipulate the view. Moreover, we already allow our videos to follow a specific syllabus—through the means of our text-to-molecumentary process. The teacher was interested in using our framework to author her own story, possibly using the scene hierarchy interface. She was particularly drawn by the text-to-molecumentary scenario. According to her, tech-savvy teachers would love to author, or at least adjust, a script of a scientific video.

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 to determine 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. The biology teacher also made this point and noted that, in scenarios when several elements interact, she would like to have further options to control how the camera shows these events. This point applies also to structures such as the membrane, for which she would rather see a cross-section to communicate the lipid bilayer. Finally, we considered only static models. Models from molecular dynamics simulations would present additional challenges.

This initial evaluation confirms people's desire for better methods in science communication. Answering questions regarding specific design decisions or alternatives, potential interaction designs, and related usability issues would require a larger-scale experiment, which we consider out of the scope of this initial technical publication but plan upon adoption of our framework by a larger audience.

8 Conclusion and Future Work

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 [3]. The goal is to develop a pipeline to automate the whole

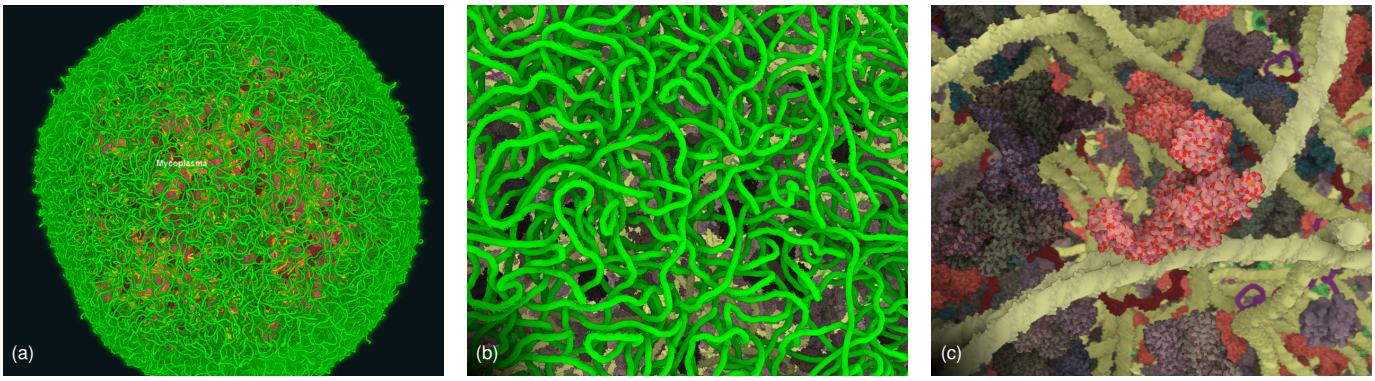

Fig. 9. The Mycoplasma model contains many more fiber instances (a). Throughout the virtual tour we visit both the strands visible from the outside view (e. g., peptides shown in (b)) as well as the insides of the bacterium pictured in (c).

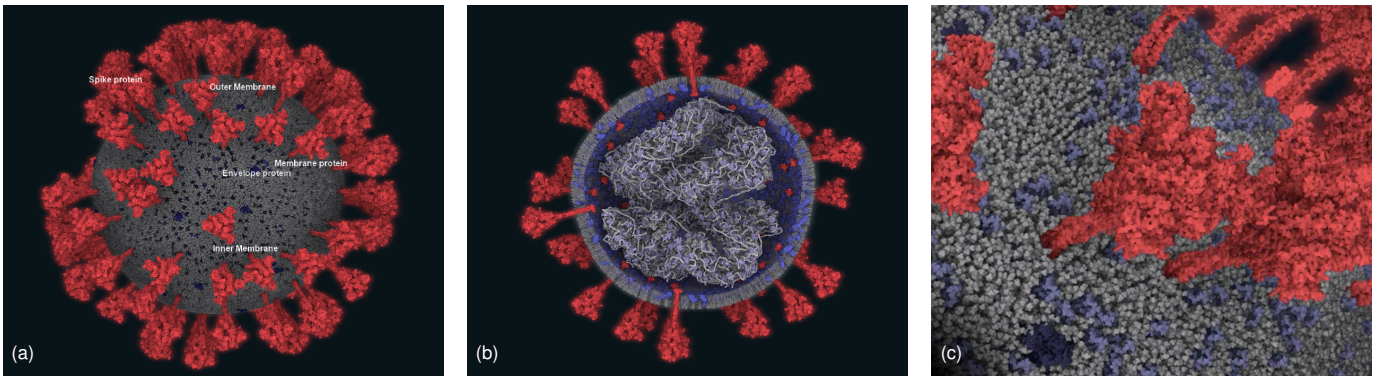

Fig. 10. The SARS-CoV-2 model shows the composition of the virus (a). We see its inside composition in an overview scene (b), and the very important structure—the spike protein—is then shown in detail in a focus scene (c).

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 communicators), 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. 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.

Many directions are possible for future work. We are interested in exploring the entire *interaction spectrum*. One end of the spectrum corresponds to fully interactive control. The other end corresponds to passive viewing without interaction. In-between, various levels of constrained navigation and guidance are worth exploring. The incorporation of artificial speech technology also suggests to exploit the opposite direction: parsing a human speech and letting the spectator's words influence the interactive experience or, in our case, the narrative of the scientific documentary.

Acknowledgments

This work was funded through the ILLUSTRARE grant by both the Austrian Science Fund (FWF): I 2953-N31, and the French National Research Agency (ANR): ANR-16-CE91-0011-01. The research was further supported by the King Abdullah University of Science and Technology (BAS/1/1680-01-01) and the ILLVISATION grant by WWTF (VRG11-010). This paper was partly written in collaboration with VRVis funded in COMET (879730) a program managed by FFG. We thank Nanographics GmbH (nanographics.at) for providing Marion.

References

[1] H. Akiba, C. Wang, and K.-L. Ma, "AniViz: A template-based animation tool for volume visualization," *IEEE Comput Graph Appl*, vol. 30, no. 5, pp. 61–71, 2010. doi: 10.1109/MCG.2009.107

[2] F. Amini, N. Henry Riche, B. Lee, C. Hurter, and P. Irani, "Understanding data videos: Looking at narrative visualization through the cinematography lens," in *Proc. CHI*. New York: ACM, 2015, pp. 1459–1468. doi: 10.1145/2702123.2702431

[3] L. Autin, M. Maritan, B. A. Barbaro, A. Gardner, A. J. Olson, M. Sanner, and D. S. Goodsell, "Mesoscope: A web-based tool for mesoscale data integration and curation," in *Proc. MolVA*. Goslar, Germany: Eurographics Assoc., 2020, pp. 23–31. doi: 10.2312/molva.20201098

[4] Å. Birkeland, S. Bruckner, A. Brambilla, and I. Viola, "Illustrative membrane clipping," *Comput Graph Forum*, vol. 31, no. 3pt1, pp. 905–914, 2012. doi: 10.1111/j.1467-8659.2012.03083.x

[5] J. Blinn, "Where am I? What am I looking at?" *IEEE Comput Graph Appl*, vol. 8, no. 4, pp. 76–81, 1988. doi: 10.1109/38.7751

[6] A. Bock, E. Axelsson, J. Costa, G. Payne, M. Acinapura, V. Trakinski, C. Emmart, C. Silva, C. Hansen, and A. Ynnerman, "OpenSpace: A system for astrographics," *IEEE Trans Vis Comput Graph*, vol. 26, no. 1, pp. 633–642, 2020. doi: 10.1109/TVCG.2019.2934259

[7] N. Burtnyk, A. Khan, G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach, "StyleCam: Interactive stylized 3D navigation using integrated spatial & temporal controls," in *Proc. UIST*. New York: ACM, 2002, pp. 101–110. doi: 10.1145/571985.572000

[8] N. Burtnyk, A. Khan, G. Fitzmaurice, and G. Kurtenbach, "ShowMotion: Camera motion based 3D design review," in *Proc. I3D*. New York: ACM, 2006, pp. 167–174. doi: 10.1145/1111411.1111442

[9] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski, "Characterizing guidance in visual analytics," *IEEE Trans Vis Comput Graph*, vol. 23, no. 1, pp. 111–120, 2017. doi: 10.1109/TVCG.2016.2598468

[10] M. Christie, R. Machap, J.-M. Normand, P. Olivier, and J. Pickering, "Virtual camera planning: A survey," in *Proc. Smart Graphics*. Berlin, Heidelberg: Springer, 2005, pp. 40–52. doi: 10.1007/11536482_4

[11] M. Christie, P. Olivier, and J.-M. Normand, "Camera control in computer graphics," *Comput Graph Forum*, vol. 27, no. 8, pp. 2197–2218, 2008. doi: 10.1111/j.1467-8659.2008.01181.x

[12] J. M. Clark and A. Paivio, "Dual coding theory and education," *Educ Psychol Rev*, vol. 3, no. 3, pp. 149–210, 1991. doi: 10.1007/BF01320076

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

[14] C. Daly, L. Clunie, and M. Ma, "From microscope to movies: 3D animations for teaching physiology," *Microscopy and Analysis*, vol. 28, no. 6, pp. 7–10, 2014.

[15] N. Elmqvist and P. Tsigas, "A taxonomy of 3D occlusion management for visualization," *IEEE Trans Vis Comput Graph*, vol. 14, no. 5, pp. 1095–1109, 2008. doi: 10.1109/TVCG.2008.59

[16] T. Fujiwara, T. Crnovrsanin, and K.-L. Ma, "Concise provenance of interactive network analysis," *Visual Informatics*, vol. 2, no. 4, pp. 213–224, 2018. doi: 10.1016/j.visinf.2018.12.002

[17] Q. Galvane, C. Lino, M. Christie, J. Fleureau, F. Servant, F.-L. Tariolle, and P. Guillotel, "Directing cinematographic drones," *ACM Trans Graph*, vol. 37, no. 3, pp. 34:1–34:18, 2018. doi: 10.1145/3181975

[18] N. Gershon and W. Page, "What storytelling can do for information visualization," *Commun ACM*, vol. 44, no. 8, pp. 31–37, 2001. doi: 10.1145/381641.381653

[19] A. Glassner, "Interactive storytelling: People, stories, and games," in *Proc. ICVS*. Berlin, Heidelberg: Springer, 2001, pp. 51–60. doi: 10.1007/3-540-45420-9_7

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

[21] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit, "From visual exploration to storytelling and back again," *Comput Graph Forum*, vol. 35, no. 3, pp. 491–500, 2016. doi: 10.1111/cgf.12925

[22] A. J. Hanson, E. A. Wernert, and S. B. Hughes, "Constrained navigation environments," in *Proc. Dagstuhl Scientific Visualization Conf.* Los Alamitos: IEEE Computer Society, 1997, pp. 95–95. doi: 10.1109/DAGSTUHL.1997.10024

[23] G. Höst, K. Palmerius, and K. Schönborn, "Nano for the public: An exploranation perspective," *IEEE Comput Graph Appl*, vol. 40, no. 2, pp. 32–42, 2020. doi: 10.1109/MCG.2020.2973120

[24] J. Hullman and N. Diakopoulos, "Visualization rhetoric: Framing effects in narrative visualization," *IEEE Trans Vis Comput Graph*, vol. 17, no. 12, pp. 2231–2240, 2011. doi: 10.1109/TVCG.2011.255

[25] J. Iwasa, "Crafting a career in molecular animation," *Mol Biol Cell*, vol. 25, no. 19, pp. 2891–2893, 2014. doi: 10.1091/mbc.e14-01-0699

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

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

[28] M. Kanehisa and S. Goto, "KEGG: Kyoto encyclopedia of genes and genomes," *Nucleic Acids Res*, vol. 28, no. 1, pp. 27–30, 2000. doi: 10.1093/nar/28.1.27

[29] R. Karpe, "A survey :On text to speech synthesis," *Int J Res Appl Sci Eng Technol*, vol. 6, no. 03, pp. 351–355, 2018. doi: 10.22214/ijraset.2018.3054

[30] R. Kosara and J. Mackinlay, "Storytelling: The next step for visualization," *IEEE Comput*, vol. 46, no. 5, pp. 44–50, 2013. doi: 10.1109/MC.2013.36

[31] D. Kouřil, T. Isenberg, B. Kozlíková, M. Meyer, E. Gröller, and I. Viola, "HyperLabels: Browsing of dense and hierarchical molecular 3D models," *IEEE Trans Vis Comput Graph*, vol. 27, no. 8, pp. 3493–3504, 2021. doi: 10.1109/TVCG.2020.2975583

[32] B. C. Kwon, F. Stoffel, D. Jäckle, B. Lee, and D. Keim, "VisJockey: Enriching data stories through orchestrated interactive visualization," in *Proc. Computation+Journalism Symp.* New York: Brown Institute for Media Innovation, 2014.

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

[34] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale, "More than telling a story: Transforming data into visually shared stories," *IEEE Comput Graph Appl*, vol. 35, no. 5, pp. 84–90, 2015. doi: 10.1109/MCG.2015.99

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

[36] I. Liao, W.-H. Hsu, and K.-L. Ma, "Storytelling via navigation: A novel approach to animation for scientific visualization," in *Proc. Smart Graphics*. Cham, Switzerland: Springer, 2014, pp. 1–14. doi: 10.1007/978-3-319-11650-1_1

[37] E. M. Lidal, H. Hauser, and I. Viola, "Geological storytelling – Graphically exploring and communicating geological sketches," in *Proc. SBIM*. Goslar, Germany: Eurographics Assoc., 2012, pp. 11–20. doi: 10.2312/SBM/SBM12/011-020

[38] C. Lino, M. Christie, F. Lamarche, G. Schofield, and P. Olivier, "A real-time cinematography system for interactive 3D environments," in *Proc. SCA*. Goslar, Germany: Eurographics Assoc., 2010, pp. 139–148. doi: 10.2312/SCA/SCA10/139-148

[39] W. E. Lorensen, "Geometric clipping using boolean textures," in *Proc. Visualization*. Los Alamitos: IEEE Comput Soc, 1993, pp. 268–274. doi: 10.1109/VISUAL.1993.398878

[40] J. Ma, I. Liao, K.-L. Ma, and J. Frazier, "Living liquid: Design and evaluation of an exploratory visualization tool for museum visitors," *IEEE Trans Vis Comput Graph*, vol. 18, no. 12, pp. 2799–2808, 2012. doi: 10.1109/TVCG.2012.244

[41] K.-L. Ma, I. Liao, J. Frazier, H. Hauser, and H. N. Kostis, "Scientific storytelling using visualization," *IEEE Comput Graph Appl*, vol. 32, no. 1, pp. 12–19, 2012. doi: 10.1109/MCG.2012.24

[42] T. M. Madhyastha and D. A. Reed, "Data sonification: Do you see what I hear?" *IEEE Softw*, vol. 12, no. 2, p. 45–56, 1995. doi: 10.1109/52.368264

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

[44] P. Mindek, D. Kouřil, J. Sorger, D. Toloudis, B. Lyons, G. Johnson, M. E. Gröller, and I. Viola, "Visualization multi-pipeline for communicating biology," *IEEE Trans Vis Comput Graph*, vol. 24, no. 1, pp. 883–892, 2017. doi: 10.1109/TVCG.2017.2744518

[45] P. Mindek, L. Čmolík, I. Viola, M. E. Gröller, and S. Bruckner, "Automatized summarization of multiplayer games," in *Proc. SCCG*. Bratislava: Comenius University, 2015, pp. 73–80. doi: 10.1145/2788539.2788549

[46] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL, USA: CRC Press, 2014. doi: 10.1201/b17511

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

[48] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Trans Graph*, vol. 36, no. 4, pp. 132:1–132:10, 2017. doi: 10.1145/3072959.3073712

[49] N. Nguyen, O. Strnad, T. Klein, D. Luo, R. Alharbi, P. Wonka, M. Maritan, P. Mindek, L. Autin, D. S. Goodsell, and I. Viola, "Modeling in the time of COVID-19: Statistical and rule-based mesoscale models," *IEEE Trans Vis Comput Graph*, vol. 27, no. 2, pp. 722–732, 2021. doi: 10.1109/TVCG.2020.3030415

[50] T. Oskam, R. W. Sumner, N. Thuerey, and M. Gross, "Visibility transition planning for dynamic camera control," in *Proc. SCA*. New York: ACM, 2009, p. 55–65. doi: 10.1145/1599470.1599478

[51] K. Perlin, "Toward interactive narrative," in *Proc. ICVS*. Berlin, Heidelberg: Springer, 2005, pp. 135–147. doi: 10.1007/11590361_16

[52] B. Preim and M. Meuschke, "A survey of medical animations," *Comput Graph*, vol. 90, pp. 145–168, 2020. doi: 10.1016/j.cag.2020.06.003

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

[54] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe, "ChartAccent: Annotation for data-driven storytelling," in *Proc. PacificVis*. Los Alamitos: IEEE Comput Soc, 2017, pp. 230–239. doi: 10.1109/PACIFICVIS.2017.8031599

[55] M. O. Riedl and R. M. Young, "From linear story generation to branching story graphs," *IEEE Comput Graph Appl*, vol. 26, no. 3, pp. 23–31, 2006. doi: 10.1109/MCG.2006.56

[56] B. Salomon, M. Garber, M. C. Lin, and D. Manocha, "Interactive navigation in complex environments using path planning," in *Proc. I3D*. New York: ACM, 2003, pp. 41–50. doi: 10.1145/641480.641491

[57] E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE Trans Vis Comput Graph*, vol. 16, no. 6, pp. 1139–1148, 2010. doi: 10.1109/TVCG.2010.179

[58] D. Siddhi, J. M. Verghese, and D. Bhavik, "Survey on various methods of text to speech synthesis," *Int J Comput Appl*, vol. 165, no. 6, pp. 26–30, 2017. doi: 10.5120/ijca2017913891

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

[60] M. Thöny, R. Schnürer, R. Sieber, L. Hurni, and R. Pajarola, "Storytelling in interactive 3D geographic visualization systems," *ISPRS Int. J. Geoinf*, vol. 7, no. 3, pp. 123:1–123:14, 2018. doi: 10.3390/ijgi7030123

[61] U. Tiede, M. Bomans, K. H. Höhne, A. Pommert, M. Riemer, T. Schiemann, R. Schubert, and W. Lierse, "A computerized three-dimensional atlas of the human skull and brain." *Am J Neuroradiol*, vol. 14, no. 3, pp. 551–559, 1993.

[62] C. Tong, R. Roberts, R. Borgo, S. Walton, R. S. Laramee, K. Wegba, A. Lu, Y. Wang, H. Qu, Q. Luo, and X. Ma, "Storytelling and visualization: An extended survey," *Inf*, vol. 9, no. 3, pp. 65:1–65:42, 2018. doi: 10.3390/info9030065

[63] J. J. van Wijk and W. A. A. Nuij, "Smooth and efficient zooming and panning," in *Proc. InfoVis*. Los Alamitos: IEEE Comput Soc, 2003, pp. 15–23. doi: 10.1109/INFVIS.2003.1249004

[64] J. Varner, "Scientific outreach: Toward effective public engagement with biological science," *BioScience*, vol. 64, no. 4, pp. 333–340, 2014. doi: 10.1093/biosci/biu021

[65] P.-P. Vázquez, T. Götzelmann, K. Hartmann, and A. Nürnberger, "An interactive 3D framework for anatomical education," *Int J Comput Assist Radiol Surg*, vol. 3, no. 6, pp. 511–524, 2008. doi: 10.1007/s11548-008-0251-4

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

[67] C. M. Wilson and S. K. Lodha, "Listen: A data sonification toolkit," in *Proc. ICAD*. Atlanta: Georgia Institute of Technology, 1996. doi: 1853/50809

[68] M. Wohlfart and H. Hauser, "Story telling for presentation in volume visualization," in *Proc. VisSym*. Goslar, Germany: Eurographics Assoc., 2007, pp. 91–98. doi: 10.2312/VisSym/EuroVis07/091-098

[69] A. Ynnerman, J. Löwgren, and L. Tibell, "Exploranation: A new science communication paradigm," *IEEE Comput Graph Appl*, vol. 38, no. 3, pp. 13–20, 2018.

[70] X. L. Zhang, "Multiscale traveling: crossing the boundary between space and scale," *Virtual Real*, vol. 13, no. 2, pp. 101–115, 2009. doi: 10.1007/s10055-009-0114-5

David Kouřil is a postdoctoral researcher at Masaryk University in Brno, Czech Republic. He received his doctoral degree from TU Wien in Vienna, Austria in April 2021. His research topic lies in scientific visualization, where he focuses on three-dimensional biological data and designs novel visualization and interaction methods that support exploration and understanding of the environments that this data represents.

Ondřej Strnad is a research scientist at KAUST, Saudi Arabia. He received his doctoral degree from Masaryk University in Brno, Czech Republic in 2014. His research interests stretch over scientific visualization, geometry algorithms and computer graphics. Recently he joined NANOVIS group at KAUST to work on technologies that deliver new visualizations and techniques regarding mesoscale biological models.

Peter Mindek is a post-doctoral researcher at TU Wien. He received his doctoral degree from TU Wien in 2015. His research interests include scientific visualization, storytelling, molecular graphics, software architecture. He co-founded Nanographics, a startup developing technology for nanovisualization.

Sarkis Halladjian has recently defended his PhD thesis on "Spatially Integrated Abstraction of Genetic Molecules" at Université Paris-Saclay, France, as a member of the Aviz research team of Inria, France. His research topic is visual abstraction in the context of multi-scale molecular visualization.

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

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

Ivan Viola is professor at King Abdullah University of Science and Technology (KAUST), Saudi Arabia. He graduated from TU Wien, Austria, in 2005 and moved for a postdoc position to the University of Bergen, Norway, where he was gradually promoted to the professor rank. In 2013 he received a WWTF grant to establish a research group at TU Wien. Viola co-founded the startup Nanographics to commercialize nanovisualization technologies.