Multi-Scale Procedural Animations of Microtubule Dynamics Based on Measured Data

Tobias Klein, Ivan Viola, Eduard Gröller, and Peter Mindek

Fig. 1: A procedurally generated, real-time rendered model of a microtubule assembly inside a cell cytoplasm.

Abstract—Biologists often use computer graphics to visualize structures, which due to physical limitations are not possible to image with a microscope. One example for such structures are microtubules, which are present in every eukaryotic cell. They are part of the cytoskeleton maintaining the shape of the cell and playing a key role in the cell division. In this paper, we propose a scientifically-accurate multi-scale procedural model of microtubule dynamics as a novel application scenario for procedural animation, which can generate visualizations of their overall shape, molecular structure, as well as animations of the dynamic behaviour of their growth and disassembly. The model is spanning from tens of micrometers down to atomic resolution. All the aspects of the model are driven by scientific data. The advantage over a traditional, manual animation approach is that when the underlying data change, for instance due to new evidence, the model can be recreated immediately. The procedural animation concept is presented in its generic form, with several novel extensions, facilitating an easy translation to other domains with emergent multi-scale behavior.

Index Terms—Procedural modeling, molecular visualization, animation, microtubules

1 Introduction

Molecular biology studies various complex structures composed of macromolecules, such as proteins or nucleic acids. These structures are highly dynamic and they carry out various tasks important for the function of a cell. However, as these structures are often smaller than the wavelength of light, their dynamics cannot be directly observed with a microscope. One way to display these structures, either for education, interdisciplinary communication, or hypothesis generation and testing, is through scientific illustration.

Biologists often display their findings about nanoscopic structures by means of computer graphics. However, 3D models of these structures have to be created first. This is a time-consuming and expensive task carried out by scientific animators who carefully study the underlying biological data and findings to manually create expressive visual representations of these phenomena. Only after the models are produced and rigged for animation, the animators can work on the overall look and feel of the animation and the storytelling aspects.

One of the structures of high interest to biologists are microtubules. They are long tubes spanning across eukaryotic cells, assembled from molecules of a protein called *tubulin*. As part of the cytoskeleton, microtubules carry out various functions, such as maintaining the cell shape, acting as channels for material transportation and as signal broadcasting system, and providing support in cell division. All of these functions are possible through a process called *dynamic instability*. It is a property of microtubules, where they constantly and in cycles grow, disassemble, and start growing again. In this way, the entire cytoskeleton is able to reshape according to the current needs of the cell.

Dynamic instability is extensively studied in biology. There are still aspects of it that are not fully understood. One of these aspects is the sequence in which the individual tubulin molecules bind together to assemble a microtubule, and the way the microtubule is disassembled. Various studies provide evidence for several theories on how this process functions. To communicate these findings, it is necessary to create illustrations, as the process itself cannot be observed with a microscope due to physical limits.

Nowadays, scientific animators manually generate dynamic 3D models representing the way how tubulins are added to a growing microtubule, or how they are removed when the microtubule shrinks. These models are created based on evidence biologists gather through various observations and experiments.

Building such models is a very time-consuming, and therefore expensive task. Since the data about the dynamic instability are not yet conclusive, it is reasonable to assume new evidence will be found and

- *Tobias Klein is with TU Wien. E-mail: tklein@cg.tuwien.ac.at.*
- *Ivan Viola is with KAUST. E-mail: ivan.viola@kaust.edu.sa.*
- *Eduard Gröller is with TU Wien and VRVis. E-mail: groeller@cg.tuwien.ac.at.*
- *Peter Mindek is with TU Wien and Nanographics GmbH. E-mail: mindek@cg.tuwien.ac.at.*

Manuscript received 31 Mar. 2019; accepted 1 Aug. 2019. Date of publication 16 Aug. 2019; date of current version 20 Oct. 2019. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2019.2934612

our understanding of the process will change. In that case, the model would have to be recreated from scratch.

To alleviate scientific animators from manually recreating microtubule models whenever new data about their growth and shrinkage are gathered, we propose a fully procedural model parameterized by measurement data. We identified several key parameters, which can be adjusted in our model according to the given theory of the microtubule dynamics, to generate a realistic visualization representing the respective theory.

To make our work extensible to other situations, where multi-scale animations of complex molecular environments are needed, we design our model within a framework for multi-scale procedural animations. The framework can be used in different application domains, where dynamic behaviour has to be modelled, and where it is essential to examine various parameter settings for the generated animations. This could be necessary, because the precise nature of the dynamics is not yet known, and hence the parameters might have to be readjusted once new evidence about the dynamics is gathered. In another example the nature of the dynamics changes according to the given context, for instance, microtubule growth might behave differently in vivo and in vitro. The ability to capture all these different characteristics within a single procedural model gives animators and biologists a powerful tool for visualizing the complex processes they are studying.

The contributions of this paper are two-fold: We introduce a novel framework for designing procedural animations in environments, where distinct behaviours on different scales interact and influence each other, for instance molecular biological processes. Subsequently, we apply this framework to design a procedural animation model on the dynamic instability of microtubules. This is an important problem studied in biology, where 3D visualizations are essential to understand the scientific data and to put them into perspective. Our procedural model can be integrated with existing real-time molecular visualization systems, so that the microtubules can be depicted in complex molecular environments, as shown in Fig. 1.

2 Related Work

Animations of molecular data are related with various techniques and processes that are fundamental to build and communicate visual narratives in a meaningful way. In a typical case, a biological model builds the foundation of the narrative. This model is assembled with a combination of data from structural biology, proteomics, and microscopy. The assembling is often realized with techniques that originate from the field of procedural modeling. Additionally, the rendering of such models implies challenges that are distinct from conventional mesh rendering. In order the explain the functionality of the showcased model, dynamics are added to the scene, which result from either animations or even simulations. In this section, we review literature corresponding to the three involved areas - procedural modeling and animation, molecular visualization, and molecular animation and storytelling.

Procedural Modeling and Animation

Models of structural biology often reveal patterns similar to other natural phenomena or man-made structures that show a high degree of geometric complexity. In visualization and computer graphics, structures that comprise a substantial amount of repetition are typically modeled in a procedural way. Early approaches procedurally generate natural phenomena, such as forests [32] and man-made structures like cities [33]. There are various ways to formulate the procedure, such as declarative modeling approaches [38], shape grammars [31], or L-systems [45]. Many approaches are even able to generate whole urban landscapes [47], or infinite cities in real-time [41, 42], a technique often used in computer games. The computational effort increases with the size and complexity of the modeled structure. A frequently used way to reduce the computation is to only generate what is currently visible in the viewing frustum of the camera [23].

Computational models of the biological meso- and nanoscale, like models of bacteria, viruses, and nucleic acids, use similar procedures for the generation. They enable users to observe the structure in atomic detail by combining data from many sources. For instance, cellPACK [18] utilizes a packing algorithm to assemble complex biological models. A more recent approach [21] extends this concept to real-time and combines it directly with the visualization.

Procedural methods for generating animations have long been used in the video-game industry for character animation, clothing [35], ocean waves [15], or clouds [36]. They simplify the process of animators to control the behavior of a model. Even for models with complex parameters, example-based approaches [2] exist that support the user in finding suitable settings. However, none of the previous approaches addresses the specifics of multi-scale models consisting of many instances, such as molecular scenes.

Molecular Visualization

Molecular visualization has a long history compared to some other data visualization branches, starting with early hand drawings of cellular environments [9] or digital drawings [48] made by computer-animation and modeling software [19]. Digital approaches often use protein building blocks from the comprehensive Protein Data Bank [1] to assemble larger biological models. There are several systems capable of rendering large datasets with up to billions of atoms [5, 27]. For instance, the cellview [25] system is a tool for illustrative multi-scale visualization of large biomolecular datasets. MegaMol [11] is capable of rendering various atomic representations not limited to biomolecules.

The structure of molecular data is complex and challenging to understand even through visualization. Its shape should be perceived without ambiguities to derive the correct information. To improve the spatial shape perception and aesthetics of molecular visualization, various rendering techniques from computer graphics have been adapted to the special characteristics of molecular data. Tarini et al. [43] have shown that the user's understanding of three-dimensional structures strongly increases with ambient occlusion. Further methods to enhance the perception include halos [12], translucency [13], depth darkening [28], and line drawings algorithms [24]. Waldin et al. [46] have shown that adjustments of the color scheme depending on the current scale level improve the capability to distinguish between different structures. Recent work by Hermosilla et al. [14] presents an illumination model that can be applied across various atom-based molecular representations.

In this paper, we utilize a framework called *Marion* [30], which incorporates a multi-pipeline approach for communicating biology. We have incorporated many essential effects to enhance the perception of the visualized structures, such as ambient occlusion, halos, depth of field, fog, and a variant of the chameleon coloring approach.

In many cases, molecular visualization shows single instances of a bacteria or viruses, where the information in between development stages is often not clear. The work of Sorger et al. [39] presents a novel approach that uses illustrative transitions and abstractions to visually transform between such stages. Molecular systems are typically visualized with various degrees of abstraction. Continuous abstraction methods [44] facilitate the creation of seamless transitions between different levels of abstraction. An extensive review of molecular visualization is presented by Kozlíková et al. [22] and Johnson and Hertig [20].

Molecular Animation and Storytelling

There is a recent shift from studying individual proteins to modeling and analyzing complex assemblies, like microtubules, up to larger systems like viruses and bacteria. These structures reside in the cellular mesoscale, which is the intermediate scale between molecular and cellular biology. To a large part, it is still an area of interpretations and assumptions. Modeling and visualization of the mesoscale is building a bridge between the molecular and the cellular scale. Animations and carefully crafted stories support the goal of building and communicating this bridge.

Goodsell et al. [10] present how mesoscale landscapes are used to construct a visual narrative. Since the process of creating an animation is tedious and time-consuming, there are first approaches [40] to reuse animation in the form of templates. The work of McGill [29] reveals the power of visualization and animation to communicate cellular and molecular structures and dynamics. It advocates to assemble visualizations into a seamless whole, the "Visual Cell". CellPAINT [7] provides a playful way for generating and exploring models of the mesoscale, which is even suitable for non-expert users. Le Muzic et al. [26] focus on temporal molecular data and propose the use of a passive agent

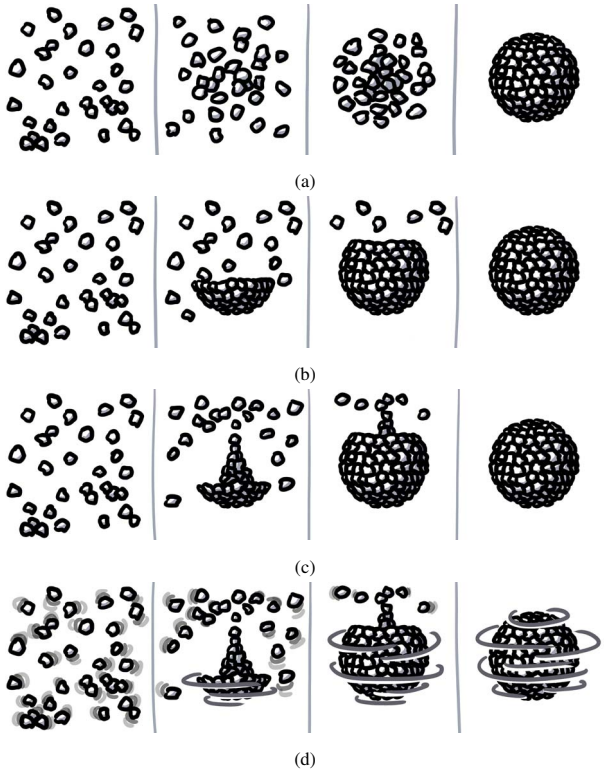


Fig. 2: (a) Four steps of a keyframed animation, where the initial state (static randomly positioned molecules) is interpolated to the target state (molecules forming a sphere). The interpolation function is independent of the molecule instance. (b) The same animation is enhanced by introducing *instance-dependent timing*. By modifying the timing of the interpolation separately for each instance, the animation is now staged. (c) An *instance-dependent transition function* is introduced. The interpolated path of each individual instance is transformed by the transition function, which can be used to model specific sequences in which the final object is built. In this case, the sphere is formed by the molecules forming a pillar, which continuously fills the sphere. (d) The same situation as in (c), but here both initial and target state are dynamic. The molecules randomly move around, while the sphere rotates around its axis.

system that controls molecular interactions according to a visual story. Recently, Iwasa [17] has reviewed different ways on how 3D animation software plays a valuable role in visualizing and communicating macromolecular structures and their dynamics.

The work of this paper goes beyond the presented approaches by tackling the problem of procedurally animating multi-scale environments.

3 Multi-scale Procedural Animations

In this paper, we describe a procedural model, which can be used to visualize complex characteristics of a biological process concerning microtubule dynamics. However, to make this approach generalizable to other applications, we first describe a conceptual framework within which our procedural dynamic model will be designed. As we show in this paper, this conceptual framework can be used to create procedural models of the dynamic behaviour in biological systems. This is an important step towards utilizing multi-scale molecular visualization not only to convey *structure*, but also *function*.

There are various approaches to create animations in molecular visualization. From the animation point of view, the most simple one is *simulation*. Sometimes, it is possible to simulate a certain environment, and the outcome of the simulation is a time-series of positions and rotations of the involved objects. The animation is created by simply displaying the time-series. However, this is not always possible, as the intrinsic properties of the environment might be unknown, or too complex, to the level where the simulation becomes infeasible. Another problem with simulations is that they cannot be easily steered by the animator to reveal certain specific aspects of the environment. Therefore, simulation might not be the most optimal choice for storytelling.

An alternative to simulation for creating molecular animations is *keyframing*. It refers to creating snapshots of the state of the environment in discrete time steps, and subsequently interpolating between them to reconstruct the state of the environment in all the missing points in time.

When using keyframing, the transformation (position and rotation) $x_{m,t}$ of an instance (in our case, a molecule) m at time t is given by:

$$x_{m,t} = mix(k_n(m), k_{n+1}(m), i) \tag{1}$$

where $k_n(m)$ is the transformation of m in the n-th keyframe and mix is the interpolation function parametrized with the interpolation parameter i. In case of simple keyframing, $i = t$, where t is the current time in the animation.

An example of a keyframed animation is shown in Fig. 2a. The figure depicts four steps of an animation. An initial keyframe (left) consisting of randomly scattered molecules and the interpolation (middle) with the target keyframe (right), where the molecules are organized into a sphere. By interpolating these keyframes, all the molecules are moved and rotated to their destination. While the animation is continuous and results in the desired outcome, it might be unrealistic in certain situations. For instance, in biology, the assemblies of larger structures are often not synchronized in a way that all molecules arrive to their destination at the same time, but rather they attach to the structure one by one.

The problem is that in the sketched situation, there are inherently two distinct scales - the behaviour of molecules is independent from the behaviour of the assembled sphere. The keyframed animation captures the environment correctly, but it cannot distinguish between the scales. During the interpolation, it applies the same rules across all the scales, thus, yielding unexpected results when applied to environments that are inherently multi-scale.

In order to support keyframe animation in multi-scale, multi-instance environments, which are ubiquitous in molecular biology, we propose the following three new concepts:

Instance-Dependent Timing A keyframed animation can be enhanced to support complex behaviour of molecular structures by modifying the interpolation function used to transition between two subsequent keyframes. The interpolation function can be made instance-dependent, as described by Sorger at al. [40]. Our approach is to assign timestamp values $t_{0,m}$ and $t_{1,m}$ to each instance m. The timestamp $t_{0,m}$ determines when the interpolation for the instance m should start, while the timestamp $t_{1,m}$ determines when it should end. Subsequently, the interpolation formula from Equation 1 is replaced by:

$$x_{m,t} = mix(k_n(m), k_{n+1}(m), i_m) \tag{2}$$

where

$$i_m = (t - t_{0,m})/(t_{1,m} - t_{0,m}) \tag{3}$$

with i_m being an instance-specific interpolation parameter.

In this way, it is possible to stage the animation for individual molecules, without the necessity of creating more keyframes than the initial and the target state. The timestamps $t_{0,m}$ and $t_{1,m}$ can be calculated procedurally for every instance. In this paper, we refer to this concept as *instance-dependent timing* (IDT) (Fig. 2b).

Instance-Dependent Transition Function While the IDT ensures that the molecules arrive to their destinations in the right order, their trajectory might still be wrong. Therefore, we introduce an *instance-dependent transition function* (IDTF) $f(x,m)$, which modifies the interpolation trajectory individually for each instance (Fig. 2c). The interpolation formula from Equation 2 becomes:

$$x_{m,t} = f(mix(k_n(m), k_{n+1}(m), i_m), m) \tag{4}$$

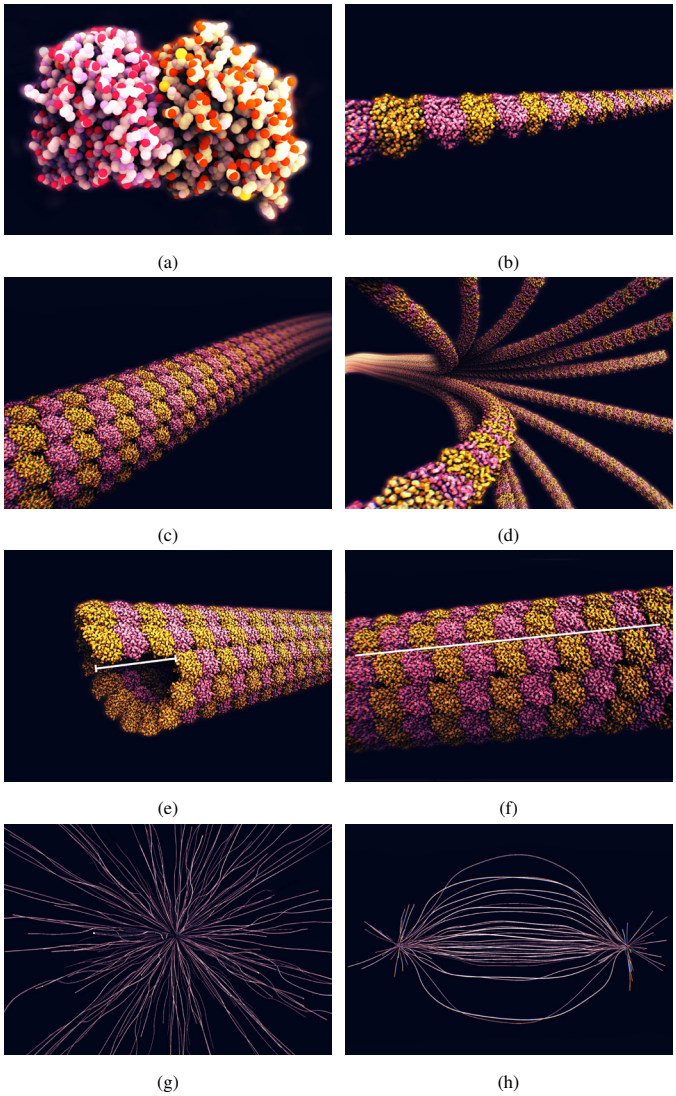


Fig. 3: (a) A tubulin dimer, consisting of two molecules - α-tubulin (purple) and β-tubulin (yellow). (b) Tubulins polymerized into a protofilament. (c) Protofilaments forming a microtubule. (d) Assembled microtubule. (e) Protofilament offset, marked with a line, forming a pseudo-helical structure of the microtubule. (f) The seam on the microtubule (white line). It is visible when the dimers are misaligned, which is caused by odd-numbered offsets. (g) A model of microtubules growing in-vitro from an isolated centrosome. (h) A model of a spindle apparatus.

Time-Varying Keyframes Fig. 2a - 2c show situations where we create an animated transition between two static states. Each of these states can be described by a simple keyframe. Now, we want to create an animated transition between two *dynamic states*. As an example, we assume an animated transition between a set of *moving* molecules and molecules organized into a rotating sphere (Fig. 2d). In this case, instead of static keyframes, we introduce *time-varying keyframes* (TVKs). A TVK is a time-dependent function $v_n(m,t)$, which specifies the transformation of the instance m at time t, in the n-th dynamic state describing our environment. In this case, v_0 describes the Brownian motion of the free-floating molecules (the initial state), and v_1 describes the molecules assembled into the sphere rotating around its axis.

To apply time-varying keyframes, the interpolation formula from Equation 4 becomes:

$$x_{m,t} = f(mix(v_n(m,t), v_{n+1}(m,t), i_m), m) \quad (5)$$

In this way, the two dynamic states are interpolated in the same way, as shown in Fig. 2d, while the dynamic behaviour on the two distinct scales is preserved. TVKs can be realized either as simulations, procedural descriptions of the dynamic states, or separate keyframe animations.

Using IDT, IDTF, and TVK, we can create keyframe animations of multi-scale, instance-based environments. This is otherwise extremely tedious with classic keyframe animation. Since we are applying interpolation to dynamic states, the velocities of the currently interpolated molecules can be distorted. In this type of animation, we are trading precision for control (as opposed to simulation, where we trade control for precision). Therefore, this type of animation is suitable for storytelling.

4 Biological Background

In this paper, we describe a multi-scale procedural model of microtubules. This particular molecular structure, found in all eukaryotic cells, is heavily researched in biology. The name microtubule was initially introduced by Slautterback [37] and their ubiquity has been revealed by Porter [34]. To communicate the structure and the function of microtubules, biologists typically use diagrams and computer generated animations. In this section, we describe the biological background necessary to understand how a procedural model of microtubules can be designed and built.

4.1 Microtubule Architecture

Microtubules are one of the building blocks of cells. They are part of the *cytoskeleton* - a complex system of protein filaments, that are carrying out various functions essential for the life of the cell. Microtubules help maintain the shape of the cell, and they act as links along which material can be transported and communication signals can be sent between different parts of cell. In eukaryotic cells, one of the crucial tasks of microtubules is to form a scaffold that helps to separate the genetic information into the daughter cells, when the cell divides. This scaffold is referred to as spindle apparatus and it only exists during the cell division.

Microtubules consist of molecules of a protein called *tubulin* (Fig. 3a). Tubulin is a dimer consisting of two tightly bound molecules - α-tubulin and β-tubulin. Tubulin molecules are dissolved in the intracellular fluid, or *cytosol*. They move around through Brownian motion caused by collisions with fast-moving fluid molecules. Upon coming into contact with each other, tubulin molecules polymerize into long chains called *protofilaments* (Fig. 3b).

Protofilaments, pushed around by the Brownian motion, tend to join into hollow tubes, which are referred to as microtubules (Fig. 3c). In eukaryotic cells, (e.g., human cells), microtubules consist of 13 protofilaments (Fig. 3d). In the most common configuration, the protofilaments

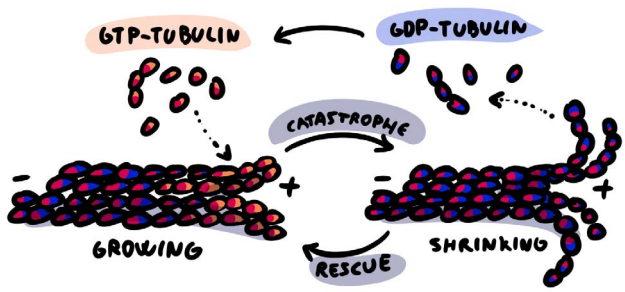


Fig. 4: Dynamic instability of microtubules. GTP-bound tubulins are added to the growing (+) end of a microtubule. GTP in the microtubule hydrolyzes into GDP. When the GDP tubulins reach all the way to the end of the microtubule, the microtubule starts to break down (catastrophe). When enough GTP tubulins are added to the shrinking end of the microtubule, the disassembly stops (rescue) and the cycle repeats. Adapted from Calligaris et al. [4].

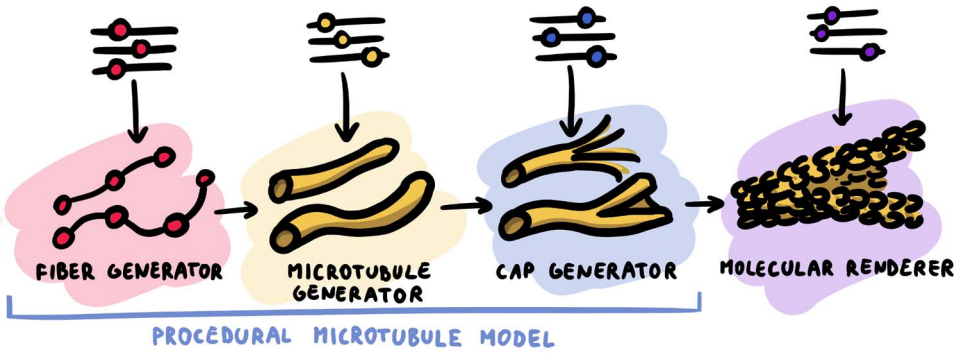


Fig. 5: Components of the pipeline for generating multi-scale visualizations of microtubules and their dynamics. First, the *fiber generator* lays out the spatial arrangement of the microtubules. This is done by generating a set of control points for each microtubule, which are then interpolated by a cubic spline. Afterwards, models of tubulin molecules are placed along each fiber in the *microtubule generator*. Subsequently, the *cap generator* displaces the tubulin molecules so that a microtubule cap is created. Finally, the model is displayed by a *molecular renderer*. Each of the modules has its own set of parameters. The blue box marks the modules belonging to the procedural microtubule model.

associate with a longitudinal offset of three tubulin monomers, forming a pseudo-helical structure of the microtubule (Fig. 3e). Since the offset length is three monomers, and a single tubulin dimer consists of two monomers, there is a seam on the microtubule. At the seam, α-tubulin and β-tubulin molecules laterally interact (Fig. 3f), as opposed to the rest of the microtubule, where α-tubulins and β-tubulins of the neighboring protofilaments are aligned.

4.2 Dynamic Instability

In order to be able to carry out all of their functions, microtubules have to be highly dynamic. They are continuously assembled and disassembled according to the current needs of the cell. The process of continuous growth and shrinkage of the microtubules is referred to as *dynamic instability*. Despite intensive study, this process has yet to be fully understood.

When assembling into a microtubule, α-tubulins are bound to β-tubulins already integrated into the microtubule. Therefore, one end of the microtubule has always the α-tubulins exposed, while the other end has the β-tubulins exposed. The end where α-tubulins are exposed is designated (-), while the one where β-tubulins are exposed is designated (+). In living eukaryotic cells, the (-) ends are anchored to *microtubule organizing centers* (*centrosomes*), while the (+) ends are continuously growing and shrinking, and in this way, exploring the space. *In vivo*, in the presence of a single, isolated centrosome, tubulin molecules polymerize into microtubules radiating away from the centrosome in all directions (Fig. 3g).

A diagram of dynamic instability is shown in Fig. 4. The free tubulin molecules dissolved in the intracellular fluid are bound to a molecule of *guanosine triphosphate* (GTP). The GTP-bound tubulins are added to the (+) end of a microtubule through their stochastic movement. Some time after joining the microtubule, the GTP molecule bound to the β-tubulin breaks down into *guanosine diphosphate* (GDP), in a process called *hydrolysis*. Protofilaments consisting of GDP-bound tubulins tend to bend outwards, as shown in Fig. 3d. It is believed that the GTP-bound tubulins at the (+) end of the microtubule hold the structure together. When the continuing hydrolysis catches up with the (+) end, the growth of the microtubule stops. This event is called *catastrophe*.

After the catastrophe, the microtubule separates into individual protofilaments, which then continue to break down into individual tubulin dimers. This rapid disassembly can be stopped when a sufficient number of GTP-bound tubulins are added to the (+) end, forming a GTP cap that will stabilize the microtubule. This event is called a *rescue*, and after it the microtubule starts to grow again.

Dynamic instability allows the microtubules to be reassembled into structures, which are needed in the given life phase of the cell. One example of this is the process of cell division. In this phase, the genetic information stored in the nucleus of the cell is copied, and the two copies need to be spatially separated. For this purpose, microtubules reassemble into a structure called *spindle apparatus* (Fig. 3h). This is achieved by moving two microtubule organizing centers into opposite sides of the cell, from where two sets of microtubules grow towards the copied genetic material. Each copy is then bound to one set of microtubules, which then act as railways for transporting each copy towards different sides of the cell. Subsequently, the cell can be divided into two new cells.

5 Multi-scale Visualization of Microtubule Dynamics

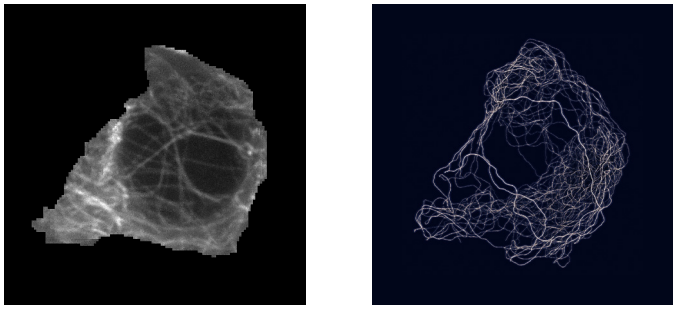
Microtubules form an inherently multi-scale structure. They span across the entire cell and can grow up to 50 micrometers. On this cellular scale, they form complex filament networks. On the molecular scale, they consist of polymerized molecules of tubulin, organized in a pseudo-helical way. Finally, on the atomic scale, processes such as hydrolysis of GTP into GDP modify the geometry of the molecular bonds, which causes the dynamic behaviour of microtubules.

Biologists study processes happening on all of these scales. To provide a robust tool for creating visualizations of their findings and theories about these processes, we propose a multi-scale procedural model of microtubules. We integrated this model into a pipeline, which allows the users to create visualizations of microtubules, either by providing the measured data as input parameters, or by manually exploring the parameter space. This makes the model useful for both domain experts and laypersons interested in microtubules. The pipeline is illustrated in Fig. 5.

5.1 Fiber Generator

The first module in the pipeline for generating the microtubule visualizations is the fiber generator. This module generates spatial curves representing individual microtubules on the cellular scale. The curves are bound to a 3D volume, representing the cytoplasm of the cell. The fiber generator creates the points of the curve by a random walk. It originates from a single point representing the microtubule organizing center. A random walk is a stochastic process, used to model polymer chains, such as microtubules, as well as other real-world phenomena. Random walks are often applied to model growth or movement, like Brownian motion.

The flexibility of polymer chains is typically described with the persistence length. It describes how orientational correlations decay along the chain. Microtubules are tubulin polymers and considered to be rigid over cellular dimensions [8]. Random walks inherently lend themselves to incorporate flexibility constraints into the generation process. At each step, the random walk chooses a new direction. To map a random walk to a given persistence length, the new direction of a step is constrained by the direction of the previous step. In the case of an unconstrained random walk, the new direction is chosen from a sphere surrounding the current point. In the constrained case, only a



(a) (b)

Fig. 6: (a) A fluorescence microscopy image of microtubules in a human-induced pluripotent stem cell. (b) Microtubule fibers generated using a segmentation mask from the scan of the same cell.

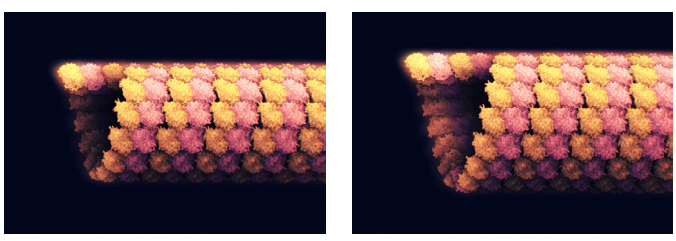
cap of the sphere is used. The center of the cap is computed from the previous direction and its size results from the given persistence length. Additional barriers limit the growth of the random walk with the result that the fiber is generated only inside of a compartment, e.g. a cell.

In Fig. 6, we show an application of the fiber generation approach that produces the curves of the microtubules. To generate a microtubule network inside a cell, we use a segmented 3D volume of a human-induced pluripotent stem cell. It has been scanned with fluorescence microscopy to restrain the generated curves within the cell cytoplasm. The segmentation mask provides the barrier for the random walk by a binary definition that reveals if the location of a step is valid, i.e. inside of the compartment, or invalid, i.e. outside of the compartment. The compartment is defined by the shape of the cell, but also by the including organelles like the nucleus. In case a random walk step is outside, the algorithm backtracks and tries again until a valid location is found.

5.2 Microtubule Generator

The microtubule generator creates the details of the microtubules on the molecular scale. It takes the curves generated by the fiber generator, and populates them with 3D models of the tubulin molecule from the PDB [1]. This is achieved by resampling the curve into uniformly spaced control points, and by generating a ring of tubulins around each point. The tubulins are aligned with the tangent of the curve at a given point. The number of tubulins along a ring and its longitudinal offset can be specified by the user. In this way, it is possible to generate microtubules of various architectures. Fig. 7 shows examples of two different microtubule architectures.

After the tubulin positions are generated, they are perturbed by a low-frequency, low-amplitude noise function to achieve a more natural look. Both frequency and amplitude should be chosen carefully, so that the solid structure of the microtubule is maintained. Fig. 8a shows a microtubule without the low-frequency noise pertubation, while Fig. 8b shows a microtubule with the noise applied.



(a) (b)

Fig. 7: Generated microtubules with two different architectures. (a) 13 protofilaments with the longitudinal offset of three monomers. (b) 16 protofilaments with the longitudinal offset of four monomers.

We use a two-dimensional *value noise* (linearly interpolated random points), where the first dimension is mapped to the length of the microtubule, while the second dimension can be mapped to time. This creates an animation with slight movements of the microtubule. Such movements naturally exist as a result of interactions with the fast-moving molecules of the surrounding medium, such as the cytosol. The advantage of value noise is that it can be easily calculated and its frequency can be specified.

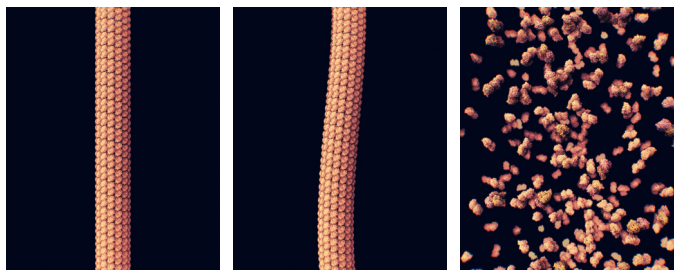
5.3 Cap Generator

The fiber generator and the microtubule generator are responsible for creating the structure of the microtubule network on the cellular and molecular scale, respectively. The third module of the pipeline, the cap generator, is responsible for creating the dynamic behaviour of individual microtubules. It generates the transition between assembled and disassembled microtubules. To make this transition realistic, we designed the cap generator according to the concept of multi-scale procedural animations introduced in Section 3. The cap generator constitutes the core contribution of the procedural microtubule model.

A microtubule is assembled from tubulin molecules, which are quickly moving around through Brownian motion. When tubulin molecules are associated to the growing microtubule, they form a structure on the growing end which we refer to as cap. One of the popular theories of how the cap looks like states that the tubulins form a flat sheet of protofilaments. The sheet quickly zips up into the tubular shape, while new tubulins are associated to the end of the cap. In this way, the microtubule grows. Similarly, when the microtubule shrinks, the tubulins are dissociated from the cap and move to the surrounding space. First, the individual protofilaments bend away from the microtubule, and then they dissolve into individual tubulins. Both growing and shrinking caps are shown in Fig. 9.

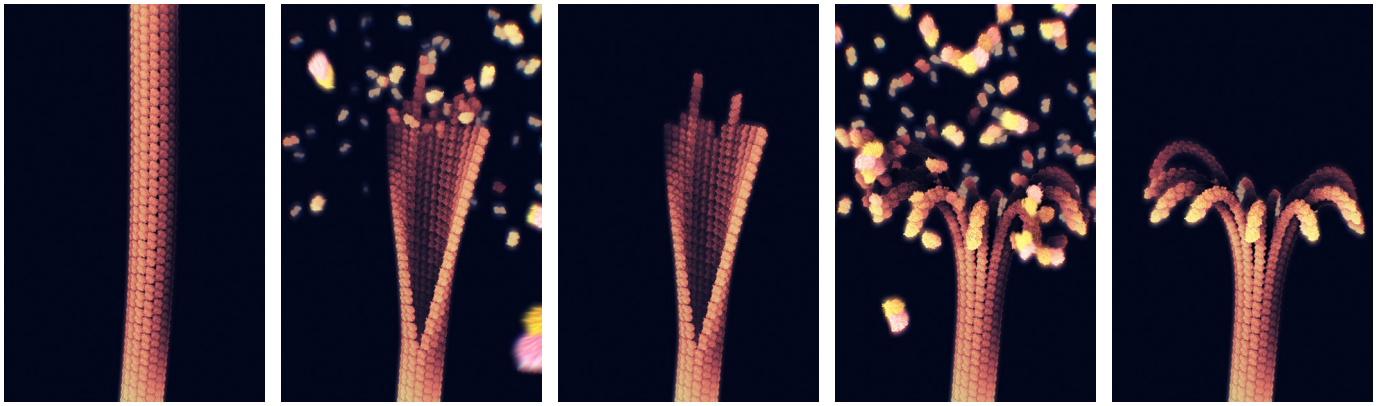
The environment of the microtubules consists of two dynamic states. One state is the Brownian motion of the free tubulin molecules. The other state is the assembled microtubule, which moves slightly around due to the collisions with the surrounding molecules. We describe the two dynamic states of the environment in a procedural way to model the assembly and disassembly process. The model of the microtubule growth and shrinking is built as interpolation between these two dynamic states, or TVKs. However, as the tubulins are associating to the microtubule and dissociating from it in a specific way, we have to use IDT and IDTF, as illustrated in Fig. 2d.

The procedural description of the assembled microtubule comes directly from the microtubule generator. Let us refer to it as v_1. We utilize it as our target TVK. The initial TVK, v_0, generates the fast stochastic movement of free tubulin molecules. Here, we perturb the positions and rotations of the molecules produced by the microtubule generator with a noise function. Similarly to the perturbation with the low-frequency noise, shown in Fig. 8b, we again use a two-dimensional value noise. While the first dimension is mapped to the tubulin position along the microtubule, the second dimension is mapped to time. The frequency in the first dimension is high enough so that the microtubule structure



(a) (b) (c)

Fig. 8: (a) A microtubule generated along a straight line. (b) A low-frequency noise applied to the microtubule to produce a slight random bending. (c) A high-frequency noise applied to the microtubule to produce the state before the microtubule assembles.



(a) (b) (c) (d) (e)

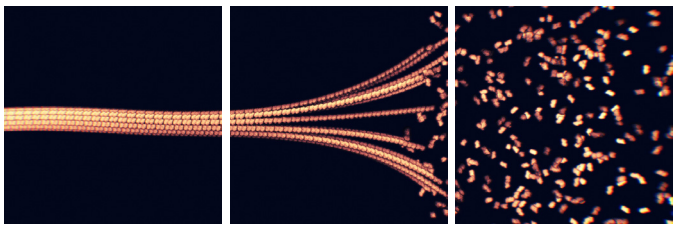
Fig. 9: (a) A microtubule produced by the microtubule generator. (b) A growing microtubule cap. (c) The growing cap with the free tubulins removed. (d) A shrinking microtubule cap. (e) The shrinking cap with the free tubulins removed.

is completely disintegrated, and the moving tubulins appear as being dissolved in the surrounding medium. The same noise perturbation is applied to the rotation of the molecules as well. The result of this transformation is shown in Fig. 8c.

In order to interpolate between v_0 and v_1, correctly visualizing the microtubule growth, we need to split a microtubule into three parts - an assembled part, a cap, a disassembled part (Fig. 10). In the assembled part, all the molecules are transformed by v_1, while in the disassembled part they are transformed by v_0. The molecules in the middle part, or the cap, currently undergo interpolation. The splitting is achieved by defining IDT for all the molecules according to their position along the microtubule. The timestamps $t_{0,m}$ and $t_{1,m}$, as defined in Equation 3, are procedurally set in a way that in the assembled part $t_{1,m} \leq t$, in the cap part $t_{0,m} \leq t < t_{1,m}$, and in the disassembled part $t_{1,m} > t$, where t is the current time. The differences between $t_{0,m}$ and $t_{1,m}$ depend on the input parameter defining the length of the cap.

The next step is to define the trajectory along which the molecules are associated to the microtubule. We do this by defining an IDTF for both the growing and the shrinking cap (Fig. 9). This is done as a geometrical transformation based on the molecule position p along the microtubule and the ID r of the protofilament it belongs to.

For the growing cap, the molecules are positioned on a flat sheet where the longitudinal position within the sheet is proportional to p, while the lateral position is proportional to r. In this way, the protofilaments are laid out next to each other. The sheet is slightly bent away from the central line of the microtubule by rotating the molecules along an axis perpendicular to the microtubule and parallel to the sheet. The axis is positioned where the cap joins the microtubule. The bending is perturbed by a low-frequency noise function to create the appearance that the growing cap is flapping due to the molecular collisions. Finally, the molecule positions within the sheet are interpolated to the positions in the assembled microtubule, to create a continuous zipping up effect.



(a) (b) (c)

Fig. 10: The interpolation between the (a) assembled and (c) disassembled states of a microtubule. The cap's middle part (b) is formed by the molecules currently undergoing the interpolation due to a specific IDT.

The IDTF for the shrinking cap is defined in a way that each molecule in the cap is rotated around the axis perpendicular to the microtubule center line and parallel to the protofilament r. This means the molecules in the same protofilament will rotate around the same axis. The amount of rotation is proportional to p. This creates the effect of individual protofilaments bending away from the microtubule - an effect characteristic for GDP-bound tubulins.

To achieve a realistic animation, it is important that the molecules are always associated only to the end of the individual protofilaments. This is achieved by selecting IDT timestamps proportionally to the position of the given molecule within the microtubule. This means that the growing or shrinking end of the microtubule is achieved by the synchronized timings of their IDTF.

The IDT timestamps define the duration of the interpolation for each molecule. Since the molecules are forced to move a certain distance in this duration, as defined by the IDTF, the speed of the molecule undergoing the interpolation might differ from the diffusion movement defined in v_0. These visual artifacts are minimized by the fact that the v_0 positions (dissolved molecules) are created by preturbing the positions in v_1 (assembled microtubule) by a noise function. This means the dissolved molecules, despite appearing randomly scattered throughout the environment, are at all times relatively close to their target position within the assembled microtubule. The molecules do not have to travel large distances during the interpolation, thus reducing visual artifacts caused by it.

With the IDT, IDTF, and TVKs defined, we can animate the growing and the shrinking of a microtubule by remapping the current time t to the microtubule length (Fig. 12). The IDTF for the growing or shrinking cap is based on whether the graph of the length over time is ascending or descending at that point. To avoid sudden changes in the shape of the cap, we can interpolate between both IDTFs when growing changes to shrinking and vice versa, as shown in Fig. 11.

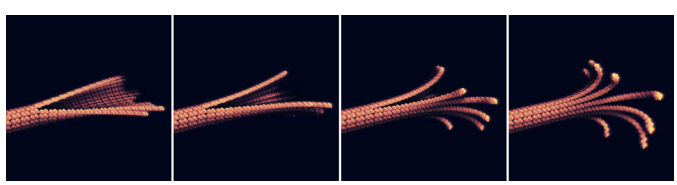


Fig. 11: Four frames of the transition between the growing and the shrinking cap of the microtubule. This is achieved by interpolating two distinct IDTFs.

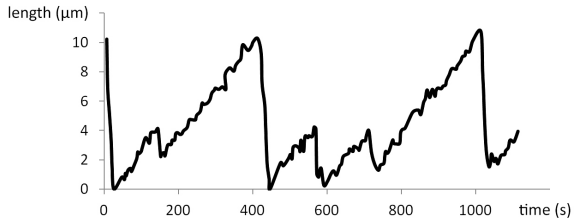


Fig. 12: Microtubule length over time. In our animation, this data is used to drive the dynamic instability. Adapted from Fygenson et al. [6].

5.4 Molecular Renderer

The last module in the pipeline is a molecular renderer. In this work, we use the molecular visualization framework Marion [30]. It supports a real-time visualization of large molecular scenes thanks to its level-of-detail scheme. It includes various graphics effects that can be used to enhance the perception of the displayed structures. In particular, we make use of screen-space ambient occlusion to enhance the perception of the macromolecular shapes of microtubules. We also employ depth-of-field and fog to improve the depth perception, motion blur for conveying the speed of the free floating molecules, and a bloom effect to highlight specific subunits of the microtubule for storytelling purposes.

Marion also implements an automatic multi-scale coloring scheme, which we use to distinguish α-tubulin and β-tubulin molecules with adaptive colors. The colors of individual tubulins are only different in close-ups, thus preventing a noisy appearance of the microtubule when rendered from a larger distance.

Marion supports a free exploration of the generated 3D scene, but also rendering the scene with pre-programmed camera paths. All the parameters of the microtubule model as well as the visualization can be keyframed. Therefore, our implementation can be used interactively to explore individual parameter settings, or non-interactively to produce animations about microtubules.

6 Results

Fig. 13 depicts visualizations generated by our pipeline. In Fig. 13a, the growing end of a microtubule is displayed. The image shows how the flat protofilament sheet closes up into the tubular shape of a microtubule. To distinguish the monomers, α-tubulins are colored in pink, while GTP-bound β-tubulins are colored in yellow.

The user can select how far along the microtubule GTP bound to β-tubulins has already hydrolysed into GDP. In Fig. 13b, the GDP-bound β-tubulins are colored in blue. The user can also define the length of the region, where the hydrolysis already started, but not all of the GTP is hydrolysed yet. Internally, the cap generator marks the GDP-bound tubulins, while the molecular renderer can assign different visual properties to them, such as different colors. Fig. 13c displays the disassembly of a microtubule. The GDP-bound β-tubulins are colored in blue. Since there is no GTP cap at the end of the microtubule, the protofilaments curl outwards and the microtubule collapses.

The application was tested using a computer with Intel Core i7-6700K CPU 4.00 GHz and NVIDIA GeForce GTX 1080 graphics card with 8 GB memory and consistently achieved interactive frame rates. This includes the computation of the model as well as the rendering.

7 Evaluation

To evaluate and discuss our procedural approach, we have collected informal feedback from several domain experts. One of the experts (P1) has more than 30 years of experience in structural biology and is skilled in scientific drawing and illustration. The second expert (P2) is a computational biologist and a certified medical illustrator with about 20 years of professional experience. This expert is specialized in the visual communication of molecular and cellular biology. The third expert (P3) studied biochemistry and biomedical visualization and has a professional background in medical animation and molecular simulation.

During the feedback session we have presented a sandbox system of the procedurally generated microtubules. In this system, a continuous animation of a growing or shrinking microtubule is shown, which is parameterizable through a graphical interface (see supplementary material).

The experts could freely explore the application and experiment with the parameters. They were asked several questions during the process. We first asked in which areas they would utilize such an application. We also inquired if they see the usefulness and effectiveness of the application for the biologists to communicate their findings. We posed questions concerning the general efficacy of the system to communicate the current knowledge about the structure of microtubules and the dynamic instability. Moreover, we wanted to know which potential they see for scientific hypothesis generation and verification.

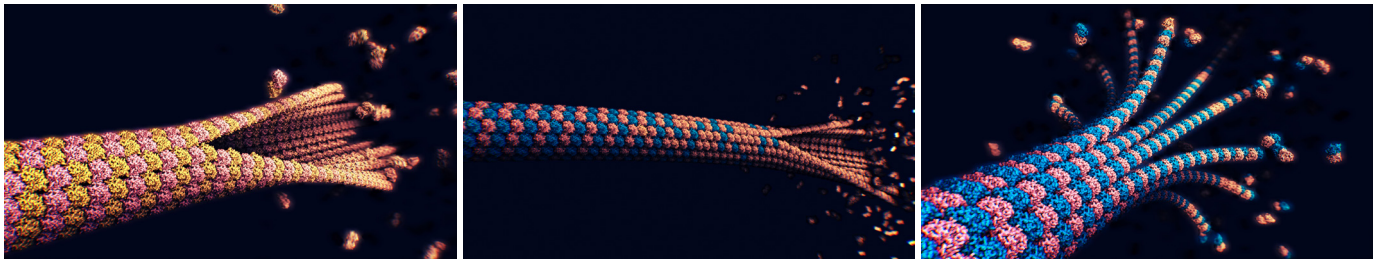
After watching the microtuble animation, expert P1 immediately mentioned that microscopy images indicate a stronger bending of the cap than in the default setting of the application. Since the application exposes the intrinsic parameters, the bending factor could be directly changed to match the current knowledge of the expert. The expert appreciated the graphical effects that create an almost "mesmerizing" animation, however, P1 mentioned that scientists communicate their findings to peers in a more simplified way in order to emphasize only the most essential features. Furthermore, the expert pointed out that the current model of the microtubule does not facilitate the generation and validation of scientific hypotheses. More vital interactions are happening that are not yet fully supported by the model. However, P1 can clearly see the usefulness of the system for communicating the microtubule processes not only to layman, but also to scientific peers. Expert P2 was "very impressed with the realtime visualization" and thinks that the "visual effects greatly contribute to the clarity of the story". The expert acknowledged that "this is the clearest and most informative description of the protofilaments merging theory and the overall process, and how the overall process relates to the catastrophe/rescue cycle, that I have seen". Expert P3 emphasized the importance of interactivity in education. He mentioned that the possibility to directly change parameters definitely improves the understanding of the displayed structures. The expert also indicated the importance of showing "reality" in education instead of very abstracted illustrations. The expert said, "this is what animators should be doing instead of animations", since it is"super tedious to build even one version of this animation". The expert would like to see further developments of the tool, so that arbitrary proteins can be added to the scene, and to have the time scale exposed to the user. For scientific publications the expert indicated that usually more schematic illustrations are preferred instead of detailed renderings.

In addition to the expert feedback, we have used the presented application to create an educational video that we submitted to the *PacificVis Visual Data Storytelling Contest*. It was evaluated by visualization and data storytelling experts and it received the award for the best storytelling video.

8 Discussion

The expert feedback demonstrates that our approach is capable of generating measurement-driven procedural animations that illustrate clear and informative aspects of the underlying biological data. The feedback made clear that the system does not fully support all required features for scientific hypothesis testing. However, the system received positive feedback concerning communicating existing knowledge and was considered superior to existing animation approaches in this aspect.

In biology, various fibrous structures play a crucial role in living cells. These structures have multiple characteristics in common, which are addressed by our approach. These structures are often synthesized by polymerization through random Brownian motion, which means that only the growing end is affected during the synthesis while the rest of the structure remains relatively static. The collisions of the building elements do not occur as frequently as when entire structures



(a) (b) (c)

Fig. 13: Visualizations generated by our pipeline. (a) shows the growing end of a microtubule, (b) microtubule with GDP-bound β-tubulins highligthed in blue, and (c) shows the disassembly of a microtubule.

are assembled at the same time. Therefore, assigning IDT and IDTF to the building elements is a powerful strategy for creating procedural animations of these processes. While we present an application of this method to model the dynamic instability of microtubules, the method could be applied to other fibrous structures as well.

Fig. 14 displays a prototype implementation of our procedural animation approach for such a polymer (DNA). In this example we show the difference between DNA in its typical helix form (Fig. 14a) and DNA in a state where part of it is unwound (Fig. 14b). The transition between the two states serves purely illustrative purposes in order to reveal the atomic structure of the individual DNA base pairs.

The goal of our procedural animation approach and most molecular animations is not to achieve the same accuracy as simulations, but rather focus on the scientific accuracy that is needed for the phenomenon to be communicated. For instance, in the implementation of our system, we have disregarded the resolving of collision for overlapping molecules. Proper collision handling would only result in minor changes in the final visualization and have insignificant influence on the particular story. Similarly, neither water nor other small molecules are present in the visualization. If a story requires proper collision handling, related work from fluid simulation [16] has demonstrated that millions of collisions of uniformly sized particles (like atoms) can be solved in real-time.

In the short term, we imagine our multi-scale procedural animation approach to be useful for generating scientifically accurate animations of chain-structured models that consist of repeating subunits. We also see our approach suitable to enhance existing simulations. Here, either coarse-grained simulations are augmented with animations of smaller structures, or vice versa, a procedural animation is enhanced with simulations of smaller structures. The approach, however, is limited to educational purposes where the result of the simulation is supposed to be communicated instead of analyzed. For the analysis of simulations more specialized tools [3] are required. In the long term, we envision our approach as a foundation for various kinds of animations of multi-scale models, such as animations of molecular machines.

As future research work, we would like to explore the capabilities of visual interfaces in order to simplify the process of describing IDTs, IDTFs, and TVKs to generate procedural animations. The work of Sorger et al. [40] already made first steps in order to reuse and simplify the description of illustrative animations with visual interfaces in the context of molecular biology. We imagine that with such an interface our approach can be used by biological illustrators and biologist with basic programming experience.

(a) (b)

Fig. 14: Interpolation between the (a) wound and (b) unwound states of DNA.

Procedural animation in multi-scale biological environments has the potential to become a widely used rapid modeling approach for authoring dynamic phenomena. As a next step, we are investigating the possible application to processes, such as DNA replication, where DNA is read in a complex sequence of operations and copied into a new strand. Another research direction is the design of visual authoring metaphors for procedural animations. Artists and biologists should be able to easily set up animations without any programming skills.

9 Conclusion

In this paper, we present a novel concept to approach the animation of dynamic processes in multi-scale, multi-instance molecular scenes. We propose an approach to design a procedural model of microtubule dynamics. This model can be used to create scientifically-accurate 3D animations of growth and shrinking of microtubules, a biological process that is intensively studied. The model can be parameterized by measured data. Examples are association and dissociation rates of tubulins when microtubules form and break down, or the length of the microtubule cap, which changes during the growing and shrinking process. In this way, biologists can create accurate visual representations of their theories and findings of how microtubule dynamic instability evolves. The model can be used for communicating scientific results, but also for education in biology.

We have evaluated the model by creating a user interface, where the parameters can be modified. We showed the software to several domain experts in biology and biological illustration and animation. The experts complimented the visual quality of the generated microtubule visualizations, as well as their suitability as both communication and education tool. Nowadays, microtubule research relies on schematic drawings and hand-made illustrations to describe the theories of this not yet completely understood process. With our procedural model, biologists can visually inspect the 3D structure of microtubules, and how different parameters would influence it. While for hypothesis testing a simulation would be needed, our procedural model gives biologists an initial idea about the implications of the given parameter values. Since the microtubule model is generated in real-time, various parameter settings can be tried quickly. While the procedural animation is presented in the context of microtubule research, it can be effectively translated into other scenarios of multi-scale emergent biological behavior.

Acknowledgments

This work was funded under the ILLVISATION grant by WWTF (VRG11-010). It is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-2019-CPF-4108 and BAS/1/1680-01-01. The paper was partly written in collaboration with the VRVis Competence Center in the scope of COMET (854174). Authors would like to thank Nanographics GmbH (nanographics.at) for providing the Marion Software Framework. Additionally, the authors wish to thank Graham Johnson and David Kouřil for the help with the implementation of the static microtubule model, and Theresia Gschwandtner for the feedback on the design of the microtubule graphics.

REFERENCES

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. doi: 10.1093/nar/28.1.235

[2] E. Brochu, T. Brochu, and N. de Freitas. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pp. 103–112. Eurographics Association, 2010.

[3] J. Byška, T. Trautner, S. M. Marques, J. Damborský, B. Kozlíková, and M. Waldner. Analysis of long molecular dynamics simulations using interactive focus+context visualization. *Computer Graphics Forum*, 2019. doi: 10.1111/cgf.13701

[4] D. Calligaris, P. Verdier-Pinard, F. Devred, C. Villard, D. Braguer, and D. Lafitte. Microtubule targeting agents: from biophysics to proteomics. *Cellular and Molecular Life Sciences*, 67(7):1089–1104, 2010. doi: 10.1007/s00018-009-0245-6

[5] M. Falk, M. Krone, and T. Ertl. Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. *Computer Graphics Forum*, 32(8):195–206, 2013. doi: 10.1111/cgf.12197

[6] D. K. Fygenson, E. Braun, and A. Libchaber. Phase diagram of microtubules. *Phys. Rev. E*, 50:1579–1588, 1994. doi: 10.1103/PhysRevE.50.1579

[7] A. Gardner, L. Autin, B. Barbaro, A. J. Olson, and D. S. Goodsell. Cellpaint: Interactive illustration of dynamic mesoscale cellular environments. *IEEE Computer Graphics and Applications*, 38(6):51–66, 2018. doi: 10.1109/MCG.2018.2877076

[8] F. Gittes, B. Mickey, J. Nettleton, and J. Howard. Flexural rigidity of microtubules and actin filaments measured from thermal fluctuations in shape. *The Journal of cell biology*, 120:923–34, 1993.

[9] D. S. Goodsell. Inside a living cell. *Trends in Biochemical Sciences*, 16:203–206, 1991. doi: 10.1016/0968-0004(91)90083-8

[10] D. S. Goodsell, M. A. Franzen, and T. Herman. From atoms to cells: Using mesoscale landscapes to construct visual narratives. *Journal of Molecular Biology*, 430(21):3954 – 3968, 2018. doi: 10.1016/j.jmb.2018.06.009

[11] S. Grottel, M. Krone, C. Müller, G. Reina, and T. Ertl. MegaMol – a prototyping framework for particle-based visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(2):201–214, 2015. doi: 10.1109/TVCG.2014.2350479

[12] P. Hermosilla, V. Guallar, A. Vinacua, and P. Vázquez. High quality illustrative effects for molecular rendering. *Computers & Graphics*, 54:113 – 120, 2016. Special Issue on CAD/Graphics 2015. doi: 10.1016/j.cag.2015.07.017

[13] P. Hermosilla, S. Maisch, P.-P. Vzquez, and T. Ropinski. Improving perception of molecular surface visualizations by incorporating translucency effects. In *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2018. doi: 10.2312/vcbm.20181244

[14] P. Hermosilla, P.-P. Vázquez Alcocer, A. Vinacua, and T. Ropinski. A general illumination model for molecular visualization. *Computer Graphics Forum*, 37:367–378, 2018. doi: 10.1111/cgf.13426

[15] D. Hinsinger, F. Neyret, and M.-P. Cani. Interactive animation of ocean waves. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pp. 161–166. ACM, New York, NY, USA, 2002. doi: 10.1145/545261.545288

[16] R. Hoetzlein. Fast fixed-radius nearest neighbors: Interactive million-particle fluids. In *GPU Technology Conference (GTC)*, 2014.

[17] J. H. Iwasa. Bringing macromolecular machinery to life using 3d animation. *Current Opinion in Structural Biology*, 31:84 – 88, 2015. Theory and simulation/Macromolecular machines and assemblies. doi: 10.1016/j.sbi.2015.03.015

[18] G. T. Johnson, L. Autin, M. Al-Alusi, D. S. Goodsell, M. F. Sanner, and A. J. Olson. cellPACK: a virtual mesoscope to model and visualize structural systems biology. *Nature methods*, 12(1):85–91, 2015.

[19] G. T. Johnson, L. Autin, D. S. Goodsell, M. F. Sanner, and A. J. Olson. ePMV embeds molecular modeling into professional animation software environments. *Structure*, 19(3):293–303, 2011.

[20] G. T. Johnson and S. Hertig. A guide to the visual analysis and communication of biomolecular structural data. *Nature Reviews Molecular Cell Biology*, 15(10):690–698, 2014.

[21] T. Klein, L. Autin, B. Kozlíková, D. S. Goodsell, A. Olson, M. E. Gröller, and I. Viola. Instant construction and visualization of crowded biological environments. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):862–872, 2018. doi: 10.1109/TVCG.2017.2744258

[22] B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum*, 2016. doi: 10.1111/cgf.13072

[23] L. Krecklau, J. Born, and L. Kobbelt. View-dependent realtime rendering of procedural facades with high geometric detail. *Computer Graphics Forum*, 32(2pt4):479–488, 2013. doi: 10.1111/cgf.12068

[24] K. Lawonn, M. Krone, T. Ertl, and B. Preim. Line integral convolution for real-time illustration of molecular surface shape and salient regions. *Computer Graphics Forum*, 2014. doi: 10.1111/cgf.12374

[25] M. Le Muzic, L. Autin, J. Parulek, and I. Viola. cellVIEW: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In *Eurographics Workshop on Visual Computing for Biology and Medicine*, pp. 61–70. EG Digital Library, The Eurographics Association, 2015.

[26] M. Le Muzic, J. Parulek, A.-K. Stavrum, and I. Viola. Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. *Computer Graphics Forum*, 33, 2014. doi: 10.1111/cgf.12370

[27] N. Lindow, D. Baum, and H.-C. Hege. Interactive rendering of materials and biological structures on atomic and nanoscopic scale. *Computer Graphics Forum*, 31(3pt4):1325–1334, 2012. doi: 10.1111/j.1467-8659.2012.03128.x

[28] T. Luft, C. Colditz, and O. Deussen. Image enhancement by unsharp masking the depth buffer. *ACM Trans. Graph.*, 25(3):1206–1213, 2006. doi: 10.1145/1141911.1142016

[29] G. McGill. Molecular movies coming to a lecture near you. *Cell*, 133(7):1127 – 1132, 2008. doi: 10.1016/j.cell.2008.06.013

[30] P. Mindek, D. Kouřil, J. Sorger, D. Toloudis, B. Lyons, G. Johnson, M. E. Gröller, and I. Viola. Visualization multi-pipeline for communicating biology. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):883–892, 2018. doi: 10.1109/TVCG.2017.2744518

[31] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. *ACM Trans. Graph.*, 25(3):614–623, 2006. doi: 10.1145/1141911.1141931

[32] F. K. Musgrave, C. E. Kolb, and R. S. Mace. The synthesis and rendering of eroded fractal terrains. *SIGGRAPH Comput. Graph.*, 23(3):41–50, 1989. doi: 10.1145/74334.74337

[33] Y. I. H. Parish and P. Müller. Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pp. 301–308. ACM, New York, NY, USA, 2001. doi: 10.1145/383259.383292

[34] K. R. Porter. *Cytoplasmic Microtubules and Their Functions*, chap. 13, pp. 308–356. John Wiley & Sons, Ltd, 2008. doi: 10.1002/9780470719442.ch13

[35] D. Rohmer, T. Popa, M.-P. Cani, S. Hahmann, and A. Sheffer. Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Trans. Graph.*, 29(6):157:1–157:8, 2010. doi: 10.1145/1882261.1866183

[36] J. Schpok, J. Simons, D. S. Ebert, and C. Hansen. A real-time cloud modeling, rendering, and animation system. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pp. 160–166. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003.

[37] D. B. Slautterback. Cytoplasmic microtubules. *The Journal of Cell Biology*, 18(2):367–388, 1963. doi: 10.1083/jcb.18.2.367

[38] R. Smelik, T. Tutenel, K. J. de Kraker, and R. Bidarra. A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics*, 35:352–363, 04 2011. doi: 10.1016/j.cag.2010.11.011

[39] J. Sorger, P. Mindek, T. Klein, G. Johnson, and I. Viola. Illustrative Transitions in Molecular Visualization via Forward and Inverse Abstraction Transform. In S. Bruckner, B. Preim, A. Vilanova, H. Hauser, A. Hennemuth, and A. Lundervold, eds., *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2016. doi: 10.2312/vcbm.20161267

[40] 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 *Proceedings of the 33rd Spring Conference on Computer Graphics*, SCCG '17, pp. 2:1–2:10. ACM, New York, NY, USA, 2017. doi: 10.1145/3154353.3154364

[41] M. Steinberger, M. Kenzel, B. Kainz, J. Müller, W. Peter, and D. Schmalstieg. Parallel generation of architecture on the GPU. *Computer Graphics Forum*, 33(2):73–82, 2014. doi: 10.1111/cgf.12312

[42] M. Steinberger, M. Kenzel, B. Kainz, P. Wonka, and D. Schmalstieg. On-

the-fly generation and rendering of infinite cities on the GPU. *Computer Graphics Forum*, 33(2):105–114, 2014. doi: 10.1111/cgf.12315

[43] M. Tarini, P. Cignoni, and C. Montani. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1237–1244, 2006. doi: 10.1109/TVCG.2006.115

[44] M. van der Zwan, W. Lueks, H. Bekker, and T. Isenberg. Illustrative molecular visualization with continuous abstraction. *Computer Graphics Forum*, 30(3):683–690, 2011. doi: 10.1111/j.1467-8659.2011.01917.x

[45] O. Šťava, B. Beneš, R. Měch, D. G. Aliaga, and P. Krištof. Inverse procedural modeling by automatic generation of L-systems. *Computer Graphics Forum*, 29(2):665–674, 2010. doi: 10.1111/j.1467-8659.2009.01636.x

[46] N. Waldin, M. Waldner, M. Le Muzic, M. E. Gröller, D. S. Goodsell, L. Autin, A. J. Olson, and I. Viola. Cuttlefish: Color mapping for dynamic multi-scale visualizations. *Computer Graphics Forum*, 2019. Preprint. doi: 10.1111/cgf.13611

[47] B. Watson, P. Müller, O. Veryovka, A. Fuller, P. Wonka, and C. Sexton. Procedural urban modeling in practice. *IEEE Computer Graphics and Applications*, 28(3):18–26, 2008. doi: 10.1109/MCG.2008.58

[48] B. G. Wilhelm, S. Mandad, S. Truckenbrodt, K. Kröhnert, C. Schäfer, B. Rammner, S. J. Koo, G. A. Claßen, M. Krauss, V. Haucke, H. Urlaub, and S. O. Rizzoli. Composition of isolated synaptic boutons reveals the amounts of vesicle trafficking proteins. *Science*, 344(6187):1023–1028, 2014. doi: 10.1126/science.1252884