論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Tensegrity representation of static elastic model for large-scale biomolecules in an interactive haptic virtual reality environment
著者(和文)	PRAMUDWIATMOKO Arif
Author(English)	Arif Pramudwiatmoko
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第11999号, 授与年月日:2021年3月26日, 学位の種別:課程博士, 審査員:山村 雅幸,小野 功,青西 亨,瀧ノ上 正浩,関嶋 政和
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11999号, Conferred date:2021/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Tensegrity representation of static elastic model for large-scale biomolecules in an interactive haptic virtual reality environment

Arif Pramudwiatmoko

A dissertation submitted to
Graduate Major in Artificial Intelligence
School of Computing
Tokyo Institute of Technology
in partial fulfillment of the requirements for
the degree of Doctor of Engineering

Supervisor: Masayuki Yamamura

December 2020

Abstract

Tensegrity representation of static elastic model for large-scale biomolecules in an interactive haptic virtual reality environment

Arif Pramudwiatmoko

Supervisor: Masayuki Yamamura

Graduate Major in Artificial Intelligence
School of Computing
Tokyo Institute of Technology

This study developed a Tensegrity representation for a static elastic object model for large-scale biomolecules in an interactive haptic VR simulation platform. A large-scale particle simulation system with full hand user interaction and haptic feedback was implemented. The tensegrity representation method creates molecular 3D objects by binding the object's particles with springs to surrounding anchors, forming an elastic tensegrity structure. The object showed the conformity of the bending shape with the classical bending equation and the viscoelastic behavior of the Kelvin (Voigt) rheological model. The parameter fitting function was defined to adjust the object's flexural rigidity to create objects with wide-range elasticity regardless of the particle formation. Together with the haptic VR system, this method forms a platform for interactive haptic VR molecular simulations.

Chapter 1 begins with an introduction to the issues underlying this research, followed by several related studies on the topic. Then the following research problems, aims, and objectives are outlined and explained: providing a VR simulation engine for large-scale biomolecular VR simulation, providing a natural user interface by hands equipped with haptic feedback, and developing a unified particle object method for creating molecular object with a wide range of elasticity values. Then the contributions are explained.

Chapter 2 reviews some of the literature as well as in more detail about existing works on microtubules, viscoelasticity, tensegrity, virtual reality for molecular simulations, haptic devices, and unified particle object models. Each existing work is reviewed, and a summary

explains gaps with the research target.

Chapter 3 begins by addressing the limitations of the existing VR molecular systems and their natural user interfaces, followed by the needs and objectives derived from these limitations. The hardware components of the natural user interface in the developed system are discussed, which consists of a hand tracking controller and a custom-built haptic rendering device. Then the development of the haptic rendering concept is explained from simple object rendering, heterogeneous surfaces, to rendering of molecular objects. Three haptic parameters are used: vibration frequency, vibration amplitude, and pressure. Tactile feedback for touching biomolecules is postulated using these three parameters. Furthermore, this chapter explains the creation of the molecular object models used in this research, followed by the implementation of haptic rendering in three different simulation frameworks, namely CHAI3D, Unity, and the DirectX 12 particle simulation system. Comparison of these hardware components with other devices is explained in a summary of the natural user interface. The next section is to find the most suitable simulation framework between CHAI3D, Unity, and DirectX 12 particle simulation system. A summary of the VR simulation system concludes the evaluation of these three frameworks. This chapter closes with discussions of the achievements, significances, problems, and limitations of the system being developed.

Chapter 4 begins by explaining the issues of existing object creation methods to describe the need for a tangible object creation method and objectives in this Tensegrity representation method. Design considerations for the object creation method is described. Then the particle simulation system for the implementation of the Tensegrity representation method is explained, followed by an explanation of the method. Next, we evaluate the Tensegrity representation object by measuring the flexural rigidity in the static force-balancing model. For the object's bending shape, we compare it with the bending shape according to the classical bending equation. We also analyze the viscoelastic behavior of objects. The results showed that the bending shape of the Tensegrity representation object was in accordance with the bending shape calculated by the classical bending equation. The viscoelastic behavior was found to be in accordance with the Kelvin (Voigt) rheological model. Then we analyze each of the object parameters to find the fitting function for the flexural rigidity. The results showed that the Tensegrity representation method can create objects with wide-range elasticity values regardless of the particle formation. The maximum spring constants to prevent overshoot of the Tensegrity representation object are then determined. Furthermore, the Tensegrity representation method is compared with other particle object methods. Then more complex mechanics are shown by the Tensegrity representation object. Lastly, the achievements,

significance, problems, and limitations of the Tensegrity representation method are discussed.

Chapter 5 begins with a discussion of the coarse-grained trade-offs in implementing the Tensegrity representation object in molecular simulations. The trade-offs are around simulation time (time course), space (size and mass scales), and force, whereas elasticity and viscosity provide additional options. Each phenomenon has a different implementation and requires specialist interpretation of the phenomenon. Then this chapter discusses the interactive live control parameters to reproduce simulations of experimental phenomena which evaluation functions are too difficult to define using existing theories. Furthermore, the interactive haptic VR simulation platform developed in this study is compared with other VR molecular systems. Lastly, the main contributions of this thesis research are outlined.

In the conclusion chapter, the main results of this thesis research are concluded. Then future works and potential difficulties with the development of the Tensegrity representation method are described. Finally, the limitations of this study are explained.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Molecular object mechanics is crucial for most of molecular systems. Microfilaments and intermediate filaments act as tension elements while microtubules act as compression elements in cytoskeleton structures. Together they form a complex network to give the cell its shape in the form of tensegrity structure at the cellular level [1][2]. Molecular filament and rod-like objects are also used in artificial systems such as molecular robotics and molecular artificial muscle [3][4]. This collection of microtubule asters in the presence of kinesin shows rapid and dynamic contractions through an energy dissipative process [4]. The emergence of the global dynamics of self-organizing massive swarms of molecular objects is one of the keys in the bottom-up fabrication of molecular robotics [5]. Mechanical properties contribute much to the emerging patterns of global dynamics, as do filament rigidity and connectivity contribute to deformation in active biopolymer networks [6]. Such phenomena are very important to observe for the development of molecular robotics.

Molecular Dynamics (MD) simulation is believed to be accurate. However, the gap between the MD simulation time course and the natural molecular phenomena is high. The time course of an MD simulation is in the order of femtoseconds whereas natural molecular phenomena should be simulated in microseconds, milliseconds, or possibly greater. A natural molecular phenomenon lasts in seconds, minutes, or even hours, depending on the phenomenon. It involves hundreds, thousands, and possibly many more molecular objects. As for the one-nanosecond phenomenon, the MD simulation in a one node in TSUBAME3.0 super computer (Intel Xeon E5-2680, two CPU of 14 cores / 28 threads 2.4 GHz, 256 Gigabytes main memories, two GPU of NVIDIA TESLA P100 for NVlink) takes about a day for protein molecules, 3 weeks for DNA origami, and one year for liposomes [7]. It can be said that simulating natural molecular phenomena using MD simulations is virtually impossible.

A coarse-grained simulation is required to simulate natural molecular phenomena consisting of large-scale biomolecular objects. Various biomolecules may be involved in the

simulation that the 3D objects created for the simulation scene need to have various values of mechanical properties as a wide range of elasticity. Some molecular objects are very flexible such as actin filaments, but some other molecular objects are very rigid such as microtubules. The different simulation time courses also affect the visual appearance of the simulation which also needs to be accommodated. Therefore, the object creation method for the simulation needs to be able to create a wide range of elasticity of objects from very flexible to very rigid. However, current methods have difficulty creating elastic objects that are very rigid.

The emergence of global dynamics in a massive swarm of molecular objects is mostly temporal. It is difficult to define an evaluation function to explain the causes for this occurrence. This lack of evaluation function makes meta-search algorithms such as genetic algorithms inapplicable. Live control of simulation parameters is a way of finding the optimal parameters governing the global dynamic behavior of molecular objects [5]. Adjusting the elasticity of objects by optimizing object parameters increases the control over the simulation to reproduce these emerging global dynamics.

The interactive haptic VR simulation platform is developed in this thesis as a tool to develop simulations of molecular phenomena in the VR environment. A powerful VR simulation engine is required to simulate large-scale biomolecules. Hand user interface with haptic feedback designed for molecular simulations adds intuition for interactive parameter optimization. To create large number of biomolecular 3D objects with various elasticities, an object creation method is needed. This thesis aims to develop an elastic object model for large-scale biomolecules in an interactive haptic VR simulation platform that we have developed.

The natural hand user interface that developed in this thesis consists of a hand tracking controller and a haptic rendering device. The Leap Motion controller is used to provide millimeter accuracy of the user's hand tracking controller [8][9]. 3D virtual hands appear in the simulation as a reflection of the user's hands being tracked by the Leap Motion controller. The virtual hands become the user's representation in the VR world to interact with molecular virtual objects. Each time the hands touch a virtual object, the simulation system sends a haptic feedback command to the haptic rendering device. The haptic rendering device is a wearable wireless device that is attached to each finger and wrist that produces touch sensation feedback with three haptic parameters: vibration frequency, vibration amplitude, and pressure strength. The haptic rendering algorithm is designed in this thesis to convert these haptic feedback parameters into a plausible tactile sense for molecular objects.

To avoid user's motion sickness, the VR system must display simulation of at least 90 frames per second (FPS). Consider the needs of the stereographic display for the left and right

eye, the number of frames rendering doubled. On the other hand, the amount of 3D mesh needed by the molecular system can reach thousands or more. This condition can overload the VR system which can greatly reduce performance. To overcome this challenge, the system must use a powerful simulation engine. A particle simulation system using multicore Central Processing Units (CPU) and multiple Graphics Processing Units (GPU) is the best candidate among several alternatives [10][11]. This system is based on a microtubule gliding assay simulation that simulates up to millions of particles using multiple GPUs [12][13][14].

Related to fulfilling system performance requirements, a coarse-grained particle modeling is required. A single 10 micrometers microtubule consists of about 109 million atoms. Therefore, molecular systems with many microtubules and other molecular objects can yield billions of atoms. Presenting each atom as a particle in the atomic model can overwhelm the system. Although this issue can be an interesting challenge for high-performance computing research, presenting objects in the coarse-grained particle model is a more viable strategy.

The 3D virtual object model developed in this thesis uses the principle of tensegrity to unite particles into an object connected with springs. This unified particle object consists of inner particles, outer particles, and springs. The inner particles represent parts of the molecular object. The outer particles are analogous to solutions that surround the object and are referred to as anchors because they bind the inner particles with springs to maintain the structural shape of the object. The formation of these particles and springs is a tensegrity structure that gives shape and mechanical rigidity to the object. This Tensegrity representation object can be created with a wide range of elasticity from very flexible to very rigid. The flexural rigidity of the object is determined by several parameters of this tensegrity representation. Therefore, tuning these parameters to match the value of the flexural rigidity of some experimental results is one of the objectives of this thesis [15].

1.2 Related Studies

The mechanical properties of molecular objects have been studied for some time with flexural rigidity being the most common properties to be measured. Researchers also acknowledge that molecular objects are surrounded by a solution that provides viscosity to resist any deformation. The viscoelasticity behavior of molecular objects arises because of the combination of their elasticity and viscosity. A lot of research studies have been done in the past, especially on measuring flexural rigidity. One of the most studied molecular objects is microtubules.

Microtubules are the most rigid parts of the cytoskeleton in the form of hollow tubes of alpha beta tubulins. They mostly work individually or in small groups; therefore, individual mechanical properties are important. The outer diameter of the microtubules is 23 to 27 nm [16] and the inner diameter is 11 to 15 nm [17]. The microtubules commonly found consist of 13 protofilaments of lateral alpha beta tubulin lattices.

Previous experimental research assumed that microtubules were homogeneous and isotropic elastic rods. However, the results of several experiments on flexural rigidity differed by up to two orders of magnitude [18]. The difference might come from various microtubule-associated proteins (MAPs) and different amounts of protofilament in the microtubules. Other differences could originate from the method chosen which determines factors such as: static / dynamic processes, analysis processes, balance and direction of the force, type of working force observed, and the number of force fulcrums [19]. Regarding these differences, this thesis is not in the position to justify them. Instead, the model we developed is intended to present whatever value is selected from the flexural rigidity of the previous results to provide an intuitive interface when showing molecular objects in a haptic VR simulation.

The viscoelasticity of a molecular object is often described by rheological models of elastic springs and viscous dashpots. Elastic springs represent elasticity which can be related to flexural rigidity. Viscous dashpots represent viscosity which is largely determined by the viscosity of the surrounding solution. The length of the molecular object affects the viscoelasticity of the object by changing the viscosity element. As in collagen molecular investigations, viscosity increases with increasing object length [20][21]. The suitable model to represent this viscoelasticity behavior is a typical Kelvin (Voigt) model with a viscous dashpot element added when the length of the object increases (Fig. 1).

Viscous dashpots not only resist any deformation, but they also cause objects to return to their original form more slowly. However, the viscous dashpots are connected in parallel with the elastic spring in the Kelvin (Voigt) model. Although it takes more time, the elastic spring will eventually push the object back to its original shape. This also happens to tensegrity objects that can withstand any deformation force and return to their original shape after the force is removed.

Tensegrity is a structure consisting of continuous tension elements and discontinuous compression elements forming a structure that stabilizes each other to maintain its shape. It was formalized by Richard Buckminster Fuller [22][23] who was inspired by the geometric model of Kenneth D. Snelson [2][24]. The structure appears like a floating object because the more visible compression elements are not connected to each other but are connected by a

network of hardly visible tension elements. This concept opposes the general structural concept which consists of compression elements that are interconnected to form a rigid structure. Nowadays, scientists are aware that this tensegrity structure does exist in nature as in living things.

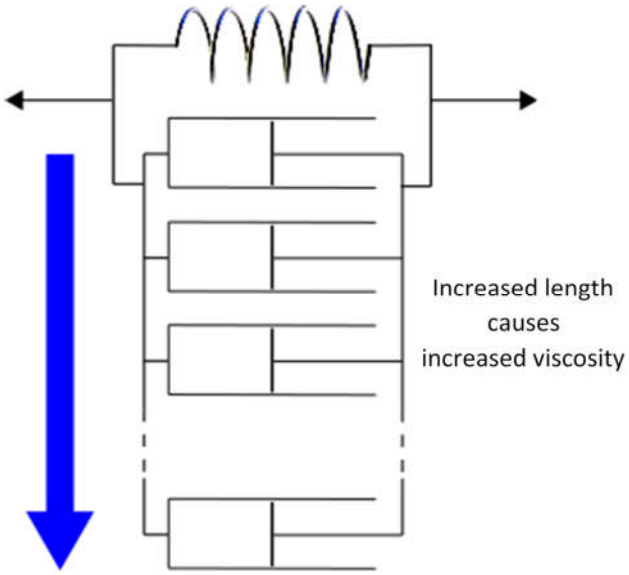


Figure 1: The Kelvin (Voigt) rheological model.
The typical viscosity model of molecular objects [15].

Tensegrity exists at all levels of living systems from the organ system level to the molecular level [1][2]. The term biotensegrity is applied to the principle of tensegrity in biological organisms [2]. The human skeletal system not only consists of bones as compression elements but also consists of muscles as tension elements. Bones will collapse without muscles that hold the balance of the skeletal system. At the cellular level, microtubules function as compression elements while microfilaments and intermediate filaments act as tension elements. Together they form a cytoskeleton system that determines the shape and movement of cells. The cytoskeleton is connected to a greater tensegrity system through integrins to connect to extracellular matrix adhesion (ECM). Any mechanical force in ECM adhesion will be transmitted to the cell nucleus [25]. ECM and microtubules have a complementary role in countering the tensional forces of cell [26] which can lead to greater scale of biotensegrity. The surrounding environment affects the strength of the microtubules to withstand the compression load. The surrounding elastic cytoskeleton is reported to reinforce microtubules in living cells, dramatically increasing the compression load that microtubules can sustain [27]. This gave us

the idea of using anchors around object's particles in our anchored spring model to mimic the surrounding environment.

VR technology enables users to easily navigate the point of view in the simulation in a 3D perspective. The ability to move and see as if the user were in the simulated world made it easy to observe objects from positions that were not possible to obtain in real laboratory experiments. This benefit has promoted the use of VR systems for education and research in molecular biology. The VR model of breast cancer cells [28] helps in cell biology education. Other systems such as Molecular Rift [29], 3D-Lab [30], ChimeraX, AltPDB, and Molecular Zoo [31] also utilize VR navigation to explore their simulations. Only some of these systems provide a natural hand user interface by supporting hand tracking controllers such as Leap Motion Controller [8][9] or Microsoft Kinect controller [32]. However, none of them simulates virtual hands and 3D objects dynamically, except MolecularZoo. MolecularZoo simulates physics but with a limited number of objects. This could be because applying a dynamic physics simulation to a large number of mesh objects is too heavy to perform using only CPU [10][11]. In addition, rendering virtual hands also causes a significant decrease in simulation performance. Therefore, a system that uses multiple GPUs is needed for this purpose. The system used in this thesis was derived from the particle simulation system used in microtubule gliding assay simulations [12][13][14]. This system utilizes a lot of GPU resources using the DirectX 12 graphics library, allowing it to simulate millions of particles using multiple GPU cards.

The use of haptic devices enhances the user experience with intuitive interaction with tactile feedback. Grounded haptic devices can perform various tactile feedbacks including force feedback for user interaction with 3D objects. Such devices are supported in the CHAI3D graphics framework. Some of the software produced by this framework are HaptiMOL ISAS [33][34], HaptiMOL ENM [35], HaptiMOL RD [36][37][38][39], and HMolDock [40]. However, due to the need for such devices to be placed on supporting objects such as tables, their mobility is very limited. On the other hand, underground haptic devices such as Go Touch VR (Go Touch VR Inc.) [41] are usually wearable devices that are highly mobile. However, such devices usually do not have force feedback to carry out complex feedback. Grabity haptic device provides hand grasping motion to the simulation and provides stiffness force feedback and weight force feedback [42]. Although this provides one directional force feedback, Grabity only represents two contact points for each hand that does not fully represent the user's hand. The haptic rendering device used in this thesis utilizes ZigBee wireless communication to make this wearable hand device highly mobile and fully wireless. It has pressure feedback for each

haptic node to simulate the strength of pressure [10][11].

1.3 Research issues

The need to simulate large-scale biomolecular 3D objects to reproduce molecular phenomena where global dynamics are emerging led to the development of an interactive haptic VR simulation platform. 3D objects created in the simulation must present mechanical properties that correspond to the behavior of the biomolecules they represent. This raises a research question; how do we create large-scale tangible molecular 3D objects that can reproduce interactions with human intervention in a VR environment?

In attempt to answer this question, a simulation platform must be established. A simulation engine that is powerful enough to simulate large-scale biomolecular 3D objects in a VR environment, an intuitive hand user interface with molecular 3D objects, and an object creation method to create tangible molecular 3D objects. The interactive haptic VR simulation platform is developed in this thesis for this.

1.4 Significance

1.4.1 Aim

This thesis aims to develop a static elastic object model for large-scale biomolecules in an interactive haptic VR simulation platform. The object model must be deployable at large scale while still meeting VR performance. A wide range of object elasticity and plausible biomolecular behavior are expected to be represented by the object model. Several goals were set to achieve this goal.

1.4.2 Objectives

The interactive haptic VR simulation platform was developed to meet the aim of this study. The three parts that make up the simulation platform are the VR simulation engine, the hand user interface, and the object creation method. To meet the aim of this study, the work in this thesis has three interrelated objectives:

- Provides a VR simulation engine for large-scale biomolecular VR simulation.
- Provides hand user interface equipped with haptic feedback for interactive VR molecular simulations.
- Develop a unified particle object method for creating molecular objects with wide range of

elasticity.

1.4.3 Contributions

This thesis uses a particle simulation system derived from microtubule gliding assay simulation [12][13][14]. In addition to the implementation of hand gesture, the haptic rendering device is also introduced with three haptic parameters: vibration frequency, vibration amplitude, and pressure strength. To implement the haptic rendering device, a haptic algorithm was developed [10][11]. A novel unified particle object called the anchored spring model was developed in this thesis. This model presents molecular objects in a particle system using the principle of tensegrity to facilitate the mechanical behavior of the objects in interactions with other 3D objects such as virtual hands [15].

Unlike most other haptic devices, the haptic rendering device combines the mobility of ungrounded haptic devices and force feedback in the form of pressure. Because touching a molecule is impossible, the feeling of touch cannot be defined. Therefore, we must postulate using molecular properties to convey tactile feedback in the three device haptic parameters. Pressure feedback only produces a sense of pressure and cannot prevent a virtual hand from penetrating a 3D object. However, because molecular objects are likely to be very soft and deformable, object penetration is permissible.

The use of the tensegrity principle in the unified particle object model is unique. This allows the object to be formed, its rigidity can be adjusted, and this can reshape to its original form after deformation. Tensegrity representation uses only spring calculations which are computationally lightweight and benefit performance. This thesis studies how to present the object model with its mechanical properties such as flexural rigidity, bending shape, and viscoelasticity behavior, while other studies have not gone that far.

The work in this thesis ultimately has two main contributions:

- Provides a tool to simulate natural phenomena with large-scale biomolecules in a VR environment with a hand user interface and haptic feedback that postulates the tactile sense when touching a biomolecular object.
- Provides a novel object creation method — the Tensegrity representation method — to create large-scale flexible 3D molecular objects with wide range of object elasticities that can be executed in VR performance.

1.5 Thesis outline

The remainder of this thesis is organized as follows. First, I review the literature related to this thesis in chapter 2. Chapter 3 describes the hand user interface, the haptic rendering concept in molecular simulation, and the VR simulation system used in this thesis to develop an interactive haptic VR simulation platform. Chapter 4 describes the Tensity representation method for creating molecular 3D objects. The discussion of the interactive haptic VR simulation platform is in chapter 5. Finally, the conclusions are presented in chapter 6. Figure 2 explains the outline of the thesis.

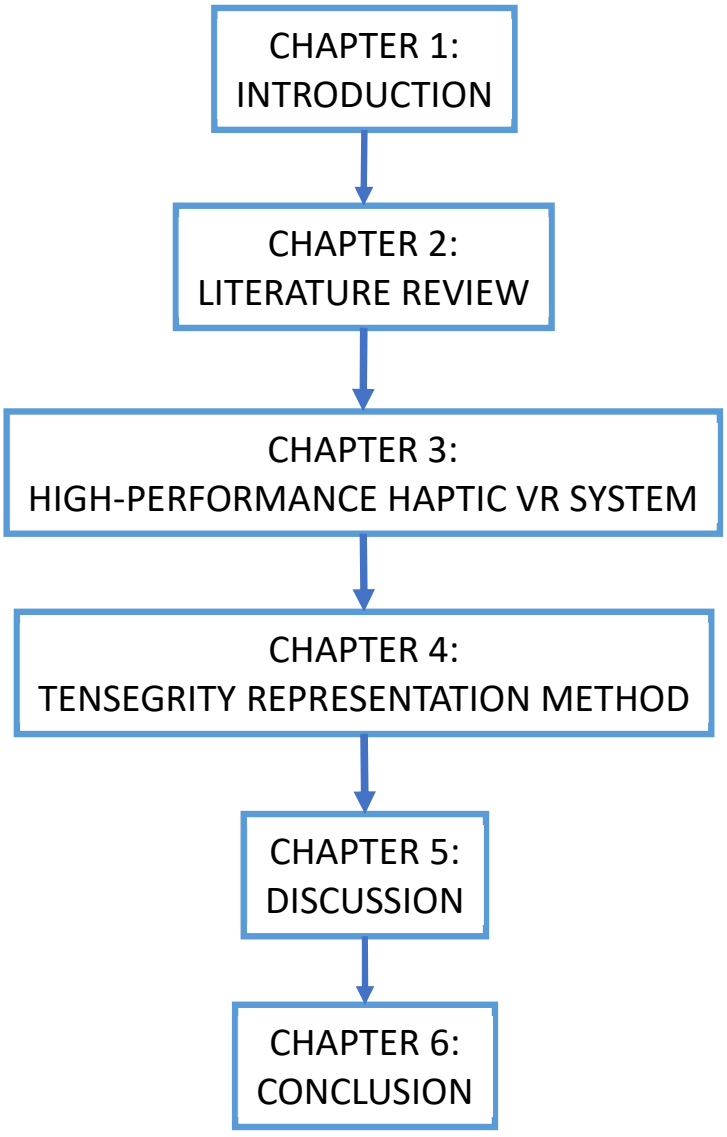


Figure 2: Thesis outline.

Chapter 2

Literature Review

In this chapter, the literature relating to this work is reviewed as background knowledge necessary to understand this thesis.

2.1 Microtubules

Microtubules were chosen to be explained to represent the molecular objects described in this thesis because they exist in most molecular systems and are often the main important objects in the system. Microtubules are the largest and most rigid cytoskeleton that play an important role in determining the shape and movement of cells. Microtubules function as tension rods that separate chromosomes in the process of cell division. Microtubules act like conveyor belts to move organelles in a cell throughout the cytoplasm. The main components of cilia and flagella are microtubules. Microtubules are also often used in artificial systems such as molecular robotics and molecular artificial muscles [3][4]. Microtubules are attached to a rod-like DNA origami object to form an aster-like structure in an artificial smooth muscle model [4].

Microtubules are hollow tube molecular polymers found in eukaryotic cells (Fig. 3). Alpha (α) and beta (β) tubulin are almost identical in shape and size and they are strongly bound to form heterodimers. Each alpha and beta tubulin monomer has a diameter of about 4 nm and a mass of 50 kDa (8,302695333 × 10-23 kg) [43]. The alpha beta tubulin dimers are components that form microtubules. The diameter of the microtubules can vary because each can have a different number of protofilaments. However, most microtubules found in cells consist of 13 protofilaments which make the outer diameter of 23 to 27 nm [16] and the inner diameter of 11 to 15 nm [17].

The length of the microtubules varies from hundreds of nanometers to tens of micrometers. The length of the microtubules can change due to the dynamic instability of the microtubules, which refers to the coexistence of the assembly (polymerization) and disassembly (depolymerization) dimers at the ends of the microtubules. One of the triggers for the growth phase and the shrinking phase of the microtubules is the concentration of alpha beta tubulin dimers in solution. Microtubules grow when the concentration of dimers is above the critical

concentration, and vice versa. Some drugs are designed to interfere with the process of polymerization or depolymerization which prevents microtubules from growing or shrinking, causing damage to the molecular system of cancer cells. Colchicine restrains microtubule polymerization, while paclitaxel (taxol) stabilizes it by preventing depolymerization. Additional substances such as taxol and MAPs also affect the rigidity of microtubules.

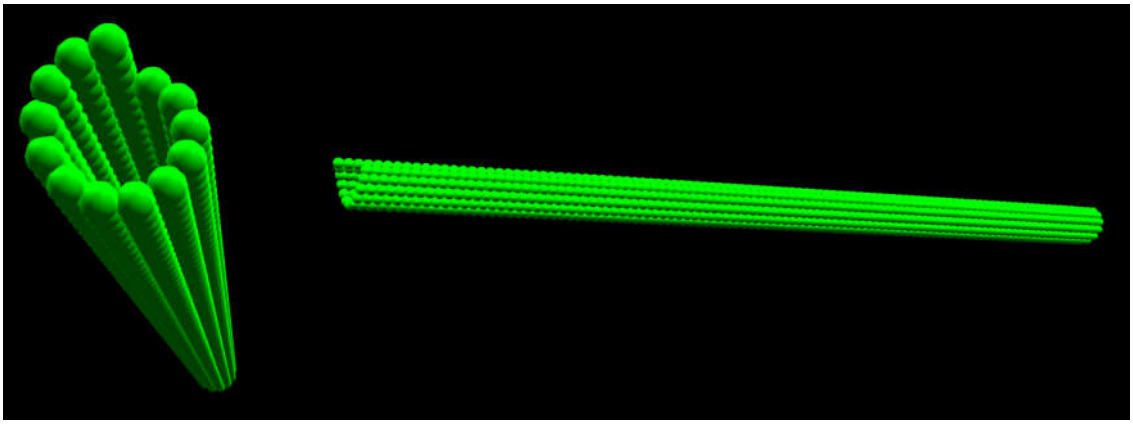


Figure 3: Microtubules are illustrated using particles.
Each particle represents an alpha or beta tubulin monomer.

Microtubules consider to be a compression element in cytoskeleton tensegrity. Their individual mechanical properties are important because they mostly work individually or in small groups. The mechanical property most relevant to their role as a compression element in tensegrity is flexural rigidity. A lot of studies have been carried out to measure the flexural rigidity of microtubules with different results up to two orders of magnitude [18]. The difference might come from various MAPs and different amounts of protofilament in the microtubules. Other differences could originate from the method chosen which determines factors such as: static / dynamic processes, analysis processes, balance and direction of the force, type of working force observed, and the number of force fulcrums [19]. Various microtubule-associated proteins (MAP) and the number of different protofilaments in microtubules may be the cause of the differences. The method chosen to measure it also contributes to differences with several distinguishing factors: static / dynamic processes, analytical processes, balance and direction of force, type of working force observed, and amount of force fulcrums. However, the fact that these experiments simplify measurements by assuming that microtubules are homogeneous and isotropic elastic rods can also cause differences to widen. In a geometric perspective, solid rod and hollow rod will deform differently under various regimes. When undergoing a low strain regime, the microtubules

bend according to the classical Euler-Bernoulli beam theory. However, above a critical strain, microtubules show a softening response due to cross-sectional flattening in the middle. Therefore, measurements carried out in low strain regimes tend to produce greater flexural rigidity values than measurements carried out in high strain regimes [44].

The methods used to measure the flexural rigidity of microtubules can be grouped into four types: the buckling force method, the relaxation method, the hydrodynamic flow method, and the thermal fluctuation method. Some parameters in this process are quite difficult to measure which can reduce precision. Because the hydrodynamic drag force is not homogeneous along the surface of the microtubules over time, methods involving this force are likely to find difficulty in accurately measuring flexural rigidity. Several uncontrolled factors are involved in the thermal fluctuation method which results in uncertainty in measurements. Kikumoto et al. [19] claims to have used the most static and direct method by using the buckling force to measure the flexural rigidity of microtubules. The experiment was carried out using two optical traps to hold both ends of the microtubule and bend it. The flexural rigidity in the experiment was calculated statically to leave only the compressive force without the influence of the hydrodynamic drag and thermal forces.

Optical traps (optical tweezers) are also used in relaxation methods. Table 1 shows some of the results of microtubule measurements using optical trap techniques. Felgner et al. [45] used optical traps in their two relaxation methods called the RELAX and WIGGLE methods. One end of the microtubule attached to the axonemes and is cemented to the coverslip, while the optical trap altered the other end. In the RELAX method, the optical trap held the other end of the microtubule and the stage was moved perpendicular to the long axis of the microtubule to make it bend in the focal plane of the microscope. The optical trap was then released for the microtubule to relax and its movement to return to the straight position was analyzed to calculate its flexural rigidity. In the WIGGLE method, the optical trap held the center of the microtubule and the stage was moved back and forth against the surrounding buffer. The wiggling movement was analyzed by calculating the flexural rigidity at the maximum deflection of the microtubule. Both methods involve the influence of the solution that create drag resistance to the movement of the microtubule. Therefore, the existence of a hydrodynamic drag force cannot be ruled out.

The optical trap technique seems to be the simplest technique to simulate. This technique does not need to simulate the motion of the solution. Basically, it moves a point of a molecular object from one position to the intended position. This technique can be used to bend molecular objects by applying a minimum of two fulcrum force points. The fulcrum points can be two

moving fulcrums (two optical traps) or one moving fulcrum and one fixed fulcrum. If measurements are made when the object stops moving (in static conditions), the force involved is only the compressive force due to the absence of friction between the object and the solution. However, if measurements are made during the movement of objects (in dynamic conditions), friction between the object and the solution occurs as a hydrodynamic drag force which cannot be ignored. Such static measurements are categorized as buckling methods. Therefore, the buckling method using optical traps is suitable to be applied in simulations.

Table 1: Several flexural rigidity measurements results using optical trap techniques [18].

Citation	Temp (ºC)	Variation	Flexural rigidity ($\times 10^{-24}$ Nm^2)	Measurement technique
[46]	37	With MAPs (10μm)	34 ± 17	Optical trap buckling
		With MAPs (30μm)	200 ± 60	Optical trap buckling
		With Taxol (5μm)	1 ± 0.65	Optical trap buckling
		With Taxol (20μm)	20 ± 6	Optical trap buckling
[45]	22-25	GDP tubulin	3.7 ± 0.8	Optical trap RELAX
		With Taxol	1 ± 0.3	Optical trap RELAX
		With MAPs	16 ± 3	Optical trap RELAX
		GDP tubulin	4.7 ± 0.4	Optical trap WIGGLE
		With Taxol	1.9 ± 0.1	Optical trap WIGGLE
		With MAPs	18 ± 3	Optical trap WIGGLE
[19]	33	GDP tubulin	7.9 ± 0.7	Optical trap buckling
		With Taxol	2.0 ± 0.8	Optical trap buckling
[47]	-	With Taxol	6.1 ± 1.3	Optical trap buckling

2.2 Viscoelasticity

Viscoelasticity is a combination of viscosity and elasticity. This is the mechanical property of objects that exhibit viscous and elastic behavior. Elastic behavior indicates the object to

change shape when a force is applied to it, but it will return to its original shape after the force is removed. On the other hand, viscous behavior resists deformation which allows objects to slow down in the deformation process and to slow down in the process of returning form. Viscoelasticity is a time-dependent behavior in which the deformed object may or may not undergo permanent deformation after the force that caused the deformation is removed. The rheological model (mechanical model) is used to describe the viscoelasticity of an object consisting of elastic and viscous elements.

An elastic object will resume to its normal form without any permanent deformation. It follows the Hooke's law which describes linear relationship between force F and corresponding extension displacement Δd as:

$$F = k\Delta d \,, \tag{1}$$

where k is a spring constant. In the rheological model (mechanical model), elasticity is modeled as an elastic spring (Fig. 4A) with Young's modulus E describing the relationship between stress σ and strain ϵ as:

$$E = \frac{\sigma}{\varepsilon} \,. \tag{2}$$

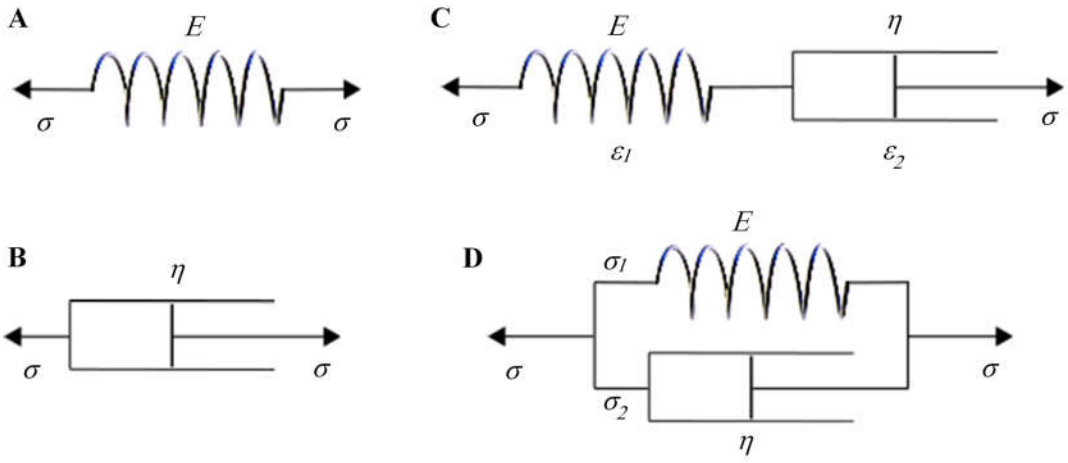


Figure 4: Basic rheological model.
(A) Linear elastic spring as an elasticity element.
(B) Linear viscous dashpot as a viscosity element.
(C) Maxwell model for viscoelasticity.
(D) Kelvin (Voigt) model for viscoelasticity.

Elasticity defines the rigidity of objects which plays an important role in most molecular systems. The elasticity or rigidity of microtubules is measured by flexural rigidity which describes the object's resistance from bending. The flexural rigidity is defined as *EI*, where *I* is the second moment of area (area moment of inertia). The cross-sectional area of the object is very influential on the value of *I* because it is a geometrical property of the area that reflects how the points are distributed with respect to the arbitrary axis. The calculation of object deflection is strongly influenced by the value of *I*.

Viscosity is the resistance of an object or material to deformation at a certain rate. It usually applies to liquid materials, but some other materials such as polymers do have these properties. Viscosity is denoted by η which is defined as:

$$\eta = \frac{\sigma}{\varepsilon}. \tag{3}$$

Viscosity is modeled as a dashpot (Fig. 4B) in the rheological model which is a piston-cylinder arrangement with viscous liquid inside. Unlike the elastic springs which return the shape of the object to its original shape, the viscous dashpot will not return to its original shape, resulting in permanent deformation of the object being modeled.

The rheological model for viscoelasticity is constructed from elastic springs and viscous dashpots in several combinations. The Maxwell model and the Kelvin model (Voigt) are two basic viscoelasticity rheological models consisting of one elastic spring and one viscous dashpot. The Maxwell model combines an elastic spring and a viscous dashpot in series (Fig. 4C), while the Kelvin model (Voigt) combines them in parallel (Fig. 4D). Other models are a combination of more than two elastic springs and viscous dashpots in a mixture between parallel and / or series formation.

In the Maxwell model, when the stress force is applied, the elastic spring will react immediately then the viscous dashpot will react slowly to the deformation. After the force is removed, the elastic spring will immediately react to recover to its original shape. However, the viscous dashpot will not change because there is no force applied to it. Therefore, objects that are described with the Maxwell model can have permanent deformation. On the other hand, the parallel combination of elastic spring and viscous dashpot in the Kelvin (Voigt) model makes it work together in pairs. Deformation of the two elements happens together. When the stress force is applied, the elastic spring will try to react immediately. However, the viscous dashpot resists deformation, making deformation happen slowly. After the force is removed,

the elastic spring will react immediately to try to recover to its original shape, while the viscous dashpot will resist the recovery effort. Because the two elements are joined, the elastic spring will force the viscous dashpot to recover. Therefore, the object described by the Kelvin model (Voigt) does not undergo permanent deformation, although shape recovery takes time.

Other models that combine more than two elastic springs and viscous dashpots may or may not have permanent deformation, it depends on the formation of the elements. If there is a viscous dashpot element that stands in series without being paralleled with an elastic spring, then the model can undergo permanent deformation because there is no element that forces the viscous dashpot to always recover its shape. If each viscous dashboard is paralleled with at least one elastic spring, the model will not experience permanent deformation because each viscous dashboard has an element that always forces it to recover its shape.

The time needed for an object to complete the deformation due to the stress force applied to it is called the retardation time. Vice versa, the time required for an object to complete the relaxation of the stress force after the force is released is called the relaxation time. Both indicate the viscosity element of the viscoelastic object. The greater the time refers to the greater viscosity. For molecular objects, the viscosity of the object is largely due to the viscosity of the surrounding solution, even though the object itself is a viscoelastic object which also has viscosity.

As reported in collagen molecular investigations, the viscosity of the molecular objects increases with increasing length [20][21]. The suitable rheological model for this molecular object is a variant of the Kelvin (Voigt) model described in Figure 1. Longer molecular objects will produce higher viscosity which is presented by adding viscous dashpot elements to the model. Higher viscosity will result in longer retardation times and longer relaxation times for objects. However, because all elements are paralleled, the object will always recover in shape.

The rheological model is useful for describing the mechanical properties of objects when subjected to simple stress forces. However, it has limitations in representing complex forces with respect to the shape of objects such as molecular filament objects. A long filament object can have multiple stress forces with various strengths and directions along the body of the object which causes it to bend differently at several points. The shape of the object is too complicated to be described by a rheological model. Therefore, this thesis focuses on the presentation of molecular objects with the tensegrity principle that is known to exist at all levels of cellular organisms.

2.3 Tensegrity

The term tensegrity comes from tensile-integrity [23] or tensional integrity [2] which is a structure consisting of continuous tension elements and discontinuous compression elements that stabilize each other to maintain its form. The structure is lightweight and can adapt to external stresses that can deform and restore its original shape. It has the impression of floating in the air because the compression elements are not connected to each other, instead they are connected by hardly visible tension elements such as strings. This is different compared to ordinary structures which consist of continuous compression elements piled together to form rigid structures. In common structural concepts, solid structures consisting of many compression elements without connections will immediately collapse. In tensegrity, this is where tension elements take on the role of connecting and stabilizing structures in unique ways. Supported by flexible structures such as the tension elements make the tensegrity structure flexible but strong enough to withstand external pressure.

The origin of tensegrity is a controversy with two names that are mostly mentioned for appreciation: Richard Buckminster Fuller [23] and Kenneth D. Snelson [48]. Fuller was inspired by a geometric model created by his art student, Snelson [2] [23][24]. Snelson continued to work with a focus on tensegrity in sculpture and aesthetic aspects to make impressive sculpture as a work of art. Meanwhile, Fuller continues to develop tensegrity using empirical experiments for applications in architecture and engineering. It is fair to say that tensegrity was formalized by Richard Buckminster Fuller [22][23] who was inspired by the geometric model Kenneth D. Snelson [2][24].

Tensegrity can be grouped into two classes: prestressed tensegrity and geodesic tensegrity [1][2]. The prestressed tensegrity structure is composed of compression elements and tension elements. The compression elements are not connected to each other but are connected and held by a network of continuous tension elements with tensional prestress (pre-existing tensile stress). The tensional prestress is very important to maintain its shape and can be adjusted to change its structural shape. On the other hand, a geodesic tensegrity structure consists of individual elements that can generate tension or resist compression depending on the external forces applied to the structure. The structure is stabilized by triangulating their structural elements to form another kind of tensional prestress. The main difference between the two classes of tensegrity is the elements that construct the structure, whereas the mathematical rules are the same [49]

The term biotensegrity refers to the tensegrity principle in biological organisms which exist

at all levels of living systems from the molecular level to organ system level [1][2]. The human skeletal system is a clear example of tensegrity at the organ system level. Bones are compression elements in tensegrity which are discontinued from each other. The skeletal system is not just a pile of bones because it will collapse without something to hold them together. If the bones are connected continuously, the skeletal system becomes rigid and the body cannot move as it is. In fact, bones are held by tension elements such as muscles, tendons, and ligaments that support the skeleton to stand strong while being flexible to move the body. The body moves its parts by changing the strength of muscle tension locally to change the position and orientation of body parts relative to the body [2].

The cytoskeleton is an example of tensegrity at the cellular level. As the largest and most rigid object, microtubules function as compression elements. Other cytoskeleton objects — microfilaments and intermediate filaments — act as tension elements. They form a cellular structure of the cytoskeleton that determines the shape and movement of eukaryotic cells. Each cytoskeleton is connected to extracellular matrix (ECM) adhesion via integrin, which can lead to greater biotensegrity scales. Mechanical forces in ECM adhesion will be transmitted to the cell nucleus, which allows cells to mechanically sense the environment [25].

Microtubules and ECM have a complementary role in countering the tensional forces of cell [26]. Microtubules gain additional strength to withstand the compression load with the surrounding environment. For example, the elastic cytoskeleton that surrounds microtubules in living cells is reported to strengthen them and dramatically increase the ability of microtubules to sustain the compression load [27]. The idea in this thesis that uses anchors around objects that bind object's particles is to imitate the surrounding environment.

2.4 Virtual reality for molecular simulations

Virtual reality (VR) technology has become a prospective technology for the future. It provides an immersive user experience by providing realistic user interactions to simulate the physical presence of users in an artificial environment. The VR system gives the user freedom to navigate observations of a simulated environment that enhance understanding of virtual objects within it. Because of the advantages in interaction with the environment presented, VR systems are widely used for education, demonstration, research, medical, etc.

VR systems come in several types that vary in the degree of immersion. Projective-based large-screen VR systems such as CAVE [50] provide walls, ceiling, and even floor that surround users with projected images arranged in a user-centered perspective. A head-mounted

display (HMD) VR system is a fully immersive VR system that consists of a stereographic display headset that covers the eyes to deliver the virtual environment to the user. Projective-based VR systems are considered less immersive because users can still see real objects in the VR room including the user's body, even though they may not be present in the virtual world. On the other hand, the HMD VR system can isolate the user's vision by only seeing virtual objects that exist in the virtual world. The increasing degree of immersion in the system significantly increases the feeling of the user's presence in the virtual world [51]; therefore, this increases the user's intuition towards objects in the simulation world.

Natural user interface (NUI) has been developed to improve intuition in a system. The user interface in NUI is direct and intuitive. Unlike the predecessor user interface that uses an artificial control device, NUI allows users to interact directly with the system such as using hands to touch or move objects. A simple example of NUI is the touch screen interface on a smartphone or touch screen monitor that allows users to click on menu items, drag icons, maximize views; instead of moving the cursor using the mouse to interact with on-screen contents. This allows user interaction to be more intuitive with the system. In a VR system, a hand tracking controller is an example of a suitable NUI that is very useful in its application.

The two most used hand tracking controller at the time of this thesis was written are the Leap Motion controller [8][9] and the Microsoft Kinect controller [32]. Both controllers provide millimeter accuracy to track the positions of user's hands and fingers. These controllers allow the user's hands to be projected in the virtual world to provide a natural user interface to interact with virtual objects. This increases the ease of the user interface and greatly increases intuition in interactions with the virtual environment. The user becomes more immersed as if he really were in a virtual world. An additional feature that can improve user intuition is the tactile sense when touching virtual objects, which is provided by haptic technology. The haptic devices will be discussed in section 2.5.

The VR model of breast cancer cell [28] is an example of a VR system in cell biology education. This system allows users to navigate through a simulated breast cancer cell world to provide intuitive learning about breast cancer by exploring deeper inside cancer in a 3D perspective. Testing shows that the system increases students' understanding of cellular processes. Molecular Rift [29] and 3D-Lab [30] are molecular viewer VR systems to assist in drug design by exploring drug molecules in a 3D perspective. Molecular Rift offers hand gesture navigation while 3D-Lab enhances it with additions such as additional collaborative web-based platforms and molecular docking methods. Similar systems are ChimeraX for analyzing molecular structure, AltPDB for atomic model collaborative discussions, and

MolecularZoo for biomolecular education [31]. Unlike Molecular Rift and 3D-Lab, these three systems do not use a hand tracking controller.

Except MolecularZoo, all VR systems mentioned are visualization systems, which do not simulate physics. Therefore, there is no physical interaction between objects and with the user's virtual hand. MolecularZoo does simulate physics by providing collisions between objects, springs, masses, and joints to present molecules with atomic connections. However, the number of simulated objects is limited [31]. The more objects or the denser the packing, the more frequent collisions occur that dramatically reduce performance and potentially fail to meet VR requirements. It uses the Nvidia PhysX engine as the default Unity game engine, which is the framework it uses. It has been reported that dynamically simulating the physics of many mesh objects is too much to be handled by CPU [10][11]. Rendering virtual hands also significantly reduces simulation performance. Physics engines that are powerful enough to do this simulation are those that use multiple GPUs.

The challenge in multiple GPUs physics engines is to make computations that are concurrent enough to be parallelized in a vast number of processors in GPUs while also minimizing data transfer between computer memory and GPU memory. The simulation system used in this thesis is derived from a previously proven particle simulation system used in the microtubule gliding assay simulation [12][13][14]. The graphical application programming interface (API) used in this system is DirectX 12, which is a powerful low-level programming that is capable to perform compute shader to utilize multiple GPU resources. This system gains benefits in performance by increasing GPU resources both quality (better GPU cards) and quantity (adding more GPU cards), enabling it to simulate millions of particles using multiple GPU cards.

2.5 Haptic devices

Haptic devices allow users to feel the sensation of touch when touching virtual objects in a simulation. These tactile sensations are generated by vibrations or force feedbacks from haptic devices designed to mimic the actual tactile feedback when touching real objects in the real world. As a user interface device, the presence of haptic devices in the simulation world can vary depending on the device, from one single haptic cursor to a pair of virtual hands. Most haptic devices present a single haptic point known as a haptic cursor that has a rotational orientation.

According to device portability, there are two types of haptic devices: grounded haptic

device and ungrounded haptic device. A grounded haptic device must be placed on a supporting structure such as a table to work properly. They usually have a haptic cursor as an interface in a virtual environment. Due to its strong footing, this device can resist any movement by generating feedback forces that opposes the user's movements. With fluctuating the feedback force, vibrating feedback can be generated and vibration parameters such as frequency and amplitude can be adjusted. Therefore, grounded haptic devices can produce various kinds of haptic feedback that is very realistic.

In terms of degrees of freedom, some haptic devices can be categorized as three degree-of-freedom (DoF) haptic devices or six degree-of-freedom haptic devices. Three degree-of-freedom (3DoF) haptic devices can move and rotate the haptic cursor in three axes (x, y, and z), but the force feedback is only in translation force without feedback of rotational force. In other words, 3DoF haptic devices have three directions of force feedback: the x, y, and z directions. On the other hand, six degrees of freedom (6DoF) haptic devices can produce translational force feedback and rotational force feedback. 6DoF haptic devices have six directions of force feedback: the x, y, and z directions in addition to rotation on the x axis (roll), the y axis (pitch), and the z axis (yaw).

Grounded haptic devices are adept at producing realistic feedback because of the ability to generate force feedback. However, such devices are heavy with complex mechanical actuators and the prices tend to be expensive. Another disadvantage is that due to the requirement to be grounded in a sturdy place, these devices are not designed as portable devices. Therefore, they are not suitable for use in a VR environment. The demands of the VR user are lightweight, high-mobility wearables that are - if possible - unnoticeable to the user. From this point of view, the ungrounded haptic devices have the advantage of being ideal haptic VR devices. Ungrounded haptic devices do not require a supporting structure to conduct haptic feedback. The only supporting structure they need is the user's body. However, due to the absence of this supporting structure, ungrounded haptic devices usually do not have force feedback. As a result, the possibility of haptic points such as virtual hands penetrating virtual objects is very high, which is usually a condition that must be avoided.

CHAI3D is a graphics framework developed primarily for creating haptic simulations. This framework supports the use of haptic devices, most of which are grounded haptic devices. It is also possible to create interfaces for various haptic devices that are not yet supported by the framework. Several research groups have used CHAI3D to develop their haptic software. HaptiMOL ISAS simulates the water accessibility of biomolecular surface using a 3DoF grounded haptic device. Using haptic cursors as an oxygen atom, this software allows users to

explore the surface of biomolecules to determine the surface area that can be accessed by the solvent [33][34]. HaptiMOL ENM uses a grounded haptic device to apply forces to individual atom in a biomolecule to deform the molecule. This software uses an elastic network model to simulate the haptic force feedback and to restructure other atoms in the molecule [35]. HaptiMOL RD simulates molecular docking of two proteins. It uses a grounded haptic device to move one protein molecule around another protein molecule to docking the molecules and simulate force feedback [36][37][38][39]. HMolDock is also a software to simulate molecular docking but uses a 6DoF haptic device with translational and rotational force feedback. This software also supports two haptic devices to move two molecules in the docking process [40]. All the CHAI3D software mentioned are not VR simulation software but their best 3D display is stereographic display.

Grabity is an ungrounded haptic device that simulates haptic feedback for hand grasping motions. This device presents two spherical haptic points in a virtual environment to allow the user to capture virtual objects. This device uses vibrotactile feedback to provide tactile sense, unidirectional brakes to create stiffness force feedback by resisting hands grasping motions, and asymmetric skin stretching to simulate weight force feedback when lifting an object [42]. It was reported that this device can simulate different object weights in a virtual environment by also providing grasping force feedback. However, the device cannot fully represent the user's hand but only represents it with two spheres that can touch and pinch objects. This device is also not fully mobile because it still relies on a cable to connect to the host computer. These limitations preclude the effort to present an intuitive simulation interface for the user even though the force feedback it can produce can be an interesting experience.

Go Touch VR (Go Touch VR Inc.) [41] is a commercial ungrounded haptic device that focuses on providing fingertip haptic feedback. The device itself does not provide a hand tracking controller but is combined with other devices that provide a hand tracking controller such as the Leap Motion controller or a more sophisticated VR glove tracking device. Go Touch VR is small, lightweight, wireless, and attached at each user's fingertips. Usually only the thumb, index finger, and middle finger use the device because it is considered sufficient for interaction. The device produces vibrations and push to the fingertips to deliver touch feedback. The device does not provide haptic feedback other than fingertips and the developer also acknowledge that it was not designed to produce delicate tactile feedback. Their intention is to focus on providing haptic feedback at three fingertips, which is most important in user interaction. Their strategy is to simply provide the feedback needed for VR simulation purposes such as providing muscle memory in touching interactions with objects in training simulations.

Based on the technology currently being developed, there will always be a trade-off between the mobility of haptic devices and the complexity of the haptic feedback that can be generated. Grounded haptic devices can produce complex and detailed haptic feedback, but they are big and heavy with low mobility. Conversely, ungrounded haptic devices can be small, lightweight, wireless, and highly mobile. However, they cannot produce haptic feedback as precisely as a grounded haptic device. While finer haptic feedback enhances user intuition, high device mobility is essential for intuitive user interaction. A wise decision is to choose a haptic device that is suitable for this purpose.

2.6 Unified particle object models

Particle simulation systems simulate all objects using a very large number of particles. Each particle is computed independently, resulting in high concurrency of computation for parallel processing. Therefore, particle simulation systems are ideal for use in large-scale simulations such as simulations of natural molecular phenomena. Since the simulation only computes particles, an object creation method is needed to integrate a group of particles into an object. A unified particle object model is created for all objects in the simulation.

The original particle simulation system in this study uses springs to integrate particles into an object [5][12][13][14]. The particles are arranged to form the shape of the object and each particle is connected by springs to the particles that surround it with a specified distance. This compound object maintains its shape by the spring forces between the particles that makes it elastic. It can be deformed by any external force, but it will try to return to its original form after the force is removed.

The structural strength of the object depends on the formation of its particles. The spring constant does affect strength although it is less influential than the formation of particles. The microtubule objects in this model are created by connecting a series using springs. Because these one-dimensional particle chains lack structural formation strength (geometric stiffness), they are very flexible. It is very difficult to make such an object rigid using that model.

The advantage of this model is its simplicity which makes computing light. Implementing large-scale simulation of biomolecular objects requires computationally efficient handling of as many particles as possible. For that reason, this model was chosen for the microtubule gliding assay simulation.

The limitations of this method are the difficulty of making the object stiff and the risk of overshoot (exploding object). Since the stiffness of objects is highly dependent on the structure

of the particle formation, objects created by this method are usually too flexible. The risk of overshoot comes from the accumulation of spring forces on the object particles. If the object's particles have too many spring connections, the accumulated force can become too large in one computation cycle causing the particles to move out of control in the next computation cycle and the object explode. Therefore, the spring constant must be set low to avoid overshoot which prevents the object to be created rigid.

As-rigid-as-possible (ARAP) simulation method with oriented particles is a unified particle object model that uses implicit Euler integration as an energy minimization with embedded oriented particles to fill the shape of the object [52]. This method can create very flexible objects that can take an extremely deformed pose and return to their original shape. The material stiffness of the object does not depend on the iteration count or the time step size. The advantages of this method are a wide range of elasticity, stability of the object elasticity, and stability of the object against the risk of overshoot.

Although ARAP methods can provide a wide range of flexibility for the objects they create, there is no sign that ARAP methods can provide enough rigidity to create nearly rigid objects. It even states that the method is not intended for accurate physics simulations but for deformable objects that visually plausible. Simulations are performed with multicore CPUs, but not GPU computing. The GPU is only used for rendering with GPU skinning. It can handle many object simulations in real-time. However, the performance is around 30 frames per second (30 Hz) in a single display which is still far from the VR performance of 90 Hz in dual displays. It was also reported that it has a performance bottleneck in collision detection events. Not using GPU computing for physics simulation may be the reason for this limited performance.

Unified particle physics method uses GPU computing for physics simulation [53]. It builds on a position-based dynamics method [54] and uses rigid shape-matching constraints [55] to maintain particle formation. It has been bundled into a GPU-based particle simulation library called NVIDIA Flex. Various types of objects can be modeled by this method: gases, liquids, rigid objects, deformable solids, and cloth. Every rigid object has a shape-matching constraint that allows for small scale elastic deformation. The deformable solids that are created are essentially rigid objects that can plastically deform when the deformation passes a threshold. Due to plastic deformation, a deformable solid does not return to its original shape after the applied force is removed. Therefore, elastic objects are only represented by rigid objects with small-scale elastic deformation.

Lovrovic and Mihajlovic [56] developed additional rigid body joints to the Unified particle

physics method to allow the application of joints between two rigid bodies. Some joint particles are used as shared possession particles between the two rigid objects. The type of joint connection is determined by the formation of joint particles. Four types of joints have been developed: stiff joint, hinge joint, ball-and-socket joint, and universal joint. The application of these joints adds to the more complex mechanisms that this method can provide.

The Unified particle physics methods that uses GPU computing for physics calculations so that it can simulate most objects in real-time. Even though the demo program was set at a maximum of 60 Hz in a single display, the reported simulation time seems small enough to make the simulation run in respectable VR performance. The performance also tends to be stable and does not experience bottleneck in collision detection events. However, physics accuracy is not the target, but only for physics that visually plausible. The fixed particle size makes it less efficient to fill objects. Large flat surfaces are not accurately represented by particles. The number of particles affects motion and deformable convergence which make this method less suitable for large objects.

2.7 Summary

Filament molecular objects are the most prevalent type of biomolecules and many studies has been devoted to measuring their mechanical properties as they are easier to observe. Flexural rigidity is measured to study the elasticity of objects. Microtubules are found in most molecular systems and are essential for systems. Therefore, filament molecular objects — especially microtubules — are suitable biomolecules to be modeled for developing object creation methods.

There are various biomolecules with different values of mechanical properties even though they are similar in shape. Even for microtubules, the variation of additional substances significantly affects rigidity. Different methods of measuring the flexural rigidity also often produce different values. More static methods have less bias by uncontrolled external forces such as hydrodynamic forces. However, static measurement methods cannot observe dynamic properties such as viscosity. Measurement of dynamic properties must be carried out in dynamic conditions, for example when an object is moved by external forces.

Viscoelasticity is a combination of viscosity and elasticity. Linear elastic springs are used as elasticity elements in the rheological model, while the linear viscous dashpots represent the viscosity elements. Biomolecules are generally a kind of Kelvin (Voigt) model in which elastic springs are parallelized with viscous dashpots. Longer biomolecular objects have higher

viscosity values; thus, they are represented by more viscous dashpots.

Tensegrity is a structure consisting of continuous tension elements and discontinuous compression elements that maintain its shape by these elements mutually stabilizing. It can withstand external pressure deforming it and return to its original shape after the pressure is removed. The term biotensity refers to the tensegrity structural principle that is present at all levels of living systems and has inspired the development of the Tensegrity representation method in this thesis.

VR and haptic technologies have been used in several molecular simulations. However, the existing VR molecular simulations are visualization systems that do not present a physics simulation. Even though some of them have physics simulations, they are limited in the number of objects they can handle. Meanwhile, the hand user interface with haptic feedback in molecular simulations has not been implemented so that the rendering of tactile feedback of touching biomolecules has not been observed.

Particle simulation system computes physics simulation in parallel which is perfect for using multicore CPUs and multiple GPUs. This system allows real-time simulation of large-scale objects. The unified particle object model which utilizes GPU compute for physics calculation is proven to have better performance and can mostly meet VR performance requirements.

The unified particle object models are suitable for creating deformable objects. However, they are designed not for accuracy in physics but for visually plausible simulations. Controlling the elasticity of objects is one of the keys to a more accurate physics simulation. Each method has a different ability to provide elasticity / rigidity to the objects it creates. Method that can produce a highly variable elasticity of objects is needed to produce a more accurate simulation.

Chapter 3

High-performance Haptic VR System

3.1 Background

This chapter describes the two components that make up the interactive haptic VR simulation platform that we develop in this thesis: the natural user interface and the VR simulation engine. The natural user interface developed for the interactive haptic VR simulation platform is a hand user interface with haptic feedback for molecular simulations. A hand tracking controller is used in conjunction with a hand haptic device. A haptic rendering concept for interacting with molecules has been proposed.

Since natural phenomena usually involve a vast of interacting molecular objects, a powerful simulation engine is needed to simulate a very large number of molecular objects. This chapter will discuss about providing a VR simulation engine that is able to meet the requirements of building a VR simulation platform for molecules. First, it is necessary to select the most suitable simulation framework for developing this platform. Then with the chosen framework, optimization is carried out in preparation for implementing the idea for the development of the intended interactive VR simulation platform.

3.1.1 Existing VR molecular systems

Most of the application of VR technology in molecular simulations is in the form of a VR molecular visualization system. Existing systems such as Molecular Rift [29], 3D-Lab [30], Caffeine [57], Molecular Dynamic Visualization (MDV) [58], VR models of breast cancer cells [28], ChimeraX, and AltPDB [31] do not perform physics simulations. MolecularZoo [31] and iMD-VR [59] do have physics simulations, but they have limitations on hand interaction with molecular objects. The number of objects the system can handle is also limited and it cannot present deformable objects. The reason for the limitation is the limited system performance because none of them use GPU computing for physics calculations.

VR models of breast cancer cells, ChimeraX, AltPDB, and iMD-VR use controller sticks to interact with the simulations, but mostly just to navigate the scene. The hand user interface is used in Molecular Rift, 3D-Lab, and MolecularZoo, but also only for navigating the scene.

None of these existing VR molecular systems provide haptic interactions. If the purpose of using VR technology is for an immersive experience as if the user is in a molecular world, the addition of a natural user interface in the form of a hand user interface with haptic feedback can certainly fill the gap in user interaction with the virtual world.

Implementing a hand user interface with haptic feedback requires a wearable glove haptic device with some haptic parameters for tactile sense. The tactile sense of touching a molecular object must be interpreted using these haptic parameters. However, no one knows the feeling of touching a molecule because of its tiny size. It takes a sensible idea to interpret molecular properties into haptic parameters so that the tactile sense of touching molecular objects in VR simulation can be realized.

Based on the description of the existing systems, there are three issues related to VR molecular and the natural user interface for the interactive haptic VR simulation platform developed in this thesis:

- Existing VR molecular systems can only simulate a limited number of objects.
- Existing VR molecular systems do not provide a natural user interface for interacting with molecular 3D objects.
- The tactile sense of touching a molecule is unknown.

3.1.2 Needs for a high-performance system

Based on the previously mention issues, to develop a natural phenomenon simulation platform with large-scale biomolecules in a VR environment requires the following:

- **High-performance VR simulation system to present large-scale biomolecules 3D objects with physics interaction.**

  A high-performance VR simulation system is needed because the computational load to process large number of objects is high, especially for 3D elastic biomolecular objects. The physics simulation of elastic 3D objects is so intense that none of the existing VR simulations mentioned in this thesis present deformable objects. Some simulation platforms exist and could potentially be used in the development of the interactive VR haptic simulation platform.

- **Natural user interface by hands and haptic feedback for interactive VR.**

  Intuitive interactions with biomolecular objects in VR simulation can be achieved with a natural user interface. The most natural way of interacting is with the hands and feeling the touch to understand the physical behavior of objects. A hand user interface with haptic

feedback is required for the interactive haptic VR simulation platform developed in this thesis.

- **Postulation of the tactile feedback of touching molecular objects.**
  Interpretation of the tactile sense when touching the molecule into the haptic feedback parameters is needed to make a plausible haptic molecule simulation. Since no one knows the feeling of touching the molecule, the tactile sense of the touch must be postulated with a sensible haptic rendering concept. The molecular properties of the object must be used to determine the haptic parameters.

3.1.3 Objectives

The objectives of this chapter are defined as stemming from the issues and needs of a high-performance haptic VR system:

- Provide a hand user interface with haptic feedback.
- Propose a haptic rendering concept of touching molecular objects.
- Provide a high-performance VR simulation system for presenting large-scale biomolecules 3D objects with physics interaction.

3.2 Hardware components

A hand tracking controller is needed to realize the hand user interface. The Leap Motion controller was chosen because it was affordable, easy to use, good software development kit (SDK) support, and quite accurate. To provide haptic feedback, a custom-built hand haptic device was developed with haptic parameters which can be used to render haptic feedback from molecular objects.

3.2.1 Hand tracking controller

The Leap Motion controller uses two monochromatic IR cameras and three infrared LEDs to obtain stereo images of the user's hand and synthesize 3D position and hand orientation data with all fingers using undisclosed calculations. The average tracking accuracy is reported at 0.7 mm and the observation area is 1-meter distance of a roughly hemispherical area. The Leap Motion controller uses two monochromatic IR cameras and three infrared LEDs to obtain a stereo image of the user's hands and synthesizes 3D position and orientation data of the hands with all fingers using undisclosed calculations. The average tracking accuracy was reported at

0.7 mm and the observation area was 1 meter to the hemispherical area. It is claimed to generate up to 200 frames per second of data according to the documentation [60], but we have experienced it generating about 115 frames per second of data.

3.2.2 Haptic rendering device

The haptic rendering device was developed by Mr. Satoru Tsutoh from the Research & Technology Group, Fuji Xerox Co., Ltd. as part of our joint research program. Our task is to implement the haptic rendering device in molecular modeling and simulation systems, which means developing algorithms for using devices in the system.

The haptic rendering device is a wireless wearable hand haptic device designed to perceive tactile sense from the user's virtual hand interaction with 3D objects in a 3D virtual environment. The haptic feedback appears in the form of vibration and pressure feedback similar to the Go Touch VR haptic device [41]. Unlike Go Touch VR, which is mounted on three fingers in each hand, the haptic rendering device consists of six vibrating parts for each hand. Five vibrating parts were placed on each finger and one was placed on the wrist. They are not completely separate but are connected through cables to share the same power source on the wrist. This is done to reduce the weight and size of the vibrating finger to provide its own battery.

The vibration feedback is generated by a linear motion vibrator (LD14-002 by Nidec Copal Electronics Inc., Tokyo Japan), which uses a solenoid coil to move a small magnet. It produces vibrations when powered by an AC signal to change the polarity of the solenoid coil. The vibration characteristics depend on the signal characteristics, which allows the frequency and amplitude to be tuned by the signal. The vibrator is attached to the fingertips in each finger vibration part, while it is placed somewhere close to the skin on the wrist vibrating part. Pressure feedback is generated by a small servo motor (KW-P0025 by K-Power Technology Co. Ltd., Guangdong China) for clamping the finger. The pressure can be adjusted by changing the rotation angle of the motor. Electrical signals for vibrators and servo motor are supplied from the embedded microcontroller — 32-bit RISC-V CPU with analog / digital interface — which is determined by haptic messages sent by the host computer.

Every vibrating part is connected to the host computer wirelessly by the ZigBee communication module (TWE-L-WX by Mono Wireless Inc., Kanagawa Japan). The host computer is installed with a Monostick USB interface to send haptic message data to the haptic rendering device. This process is illustrated in Figure 5. Each vibrating part is equipped with a

unique hexadecimal ID number to be the haptic message address of the host computer. Every time a collision event involving a virtual hand occurs that triggers the simulation to produce a haptic response, the host computer broadcasts a haptic message containing the ID number and haptic parameters to all parts of the device. Each haptic message is addressed to one vibrating part. The vibrating part with the ID number that matches the haptic message will respond to the message by generating haptic feedback according to the haptic parameters of the message.

Figure 5: Haptic rendering device communication scheme.
Computer broadcasts a command message with ID number. The vibrating part with the correct ID number executes the command for actuation [10][11].

The Leap Motion Controller is used to track the position and orientation of the hands with all the fingers. The virtual hands move according to data from the Leap Motion controller. Figure 6 shows a haptic rendering device, virtual hand, Monostick USB interface, and Leap Motion controller. Leap Motion controller can be used to track hands and fingers movements in common 3D simulations or in VR environments. For common simulations, the controller needs to be placed on a table to let the user's hands can be tracked as it moves over it. Whereas in VR simulation, the controller must be attached on the head mounted display in front of the eye position. The controller will spot the user's hand and track the hand gesture to obtain the position and orientation of all fingers. The tracking data is then sent to the host computer with a USB connection and the simulation system will present the user's hands as 3D virtual hands.

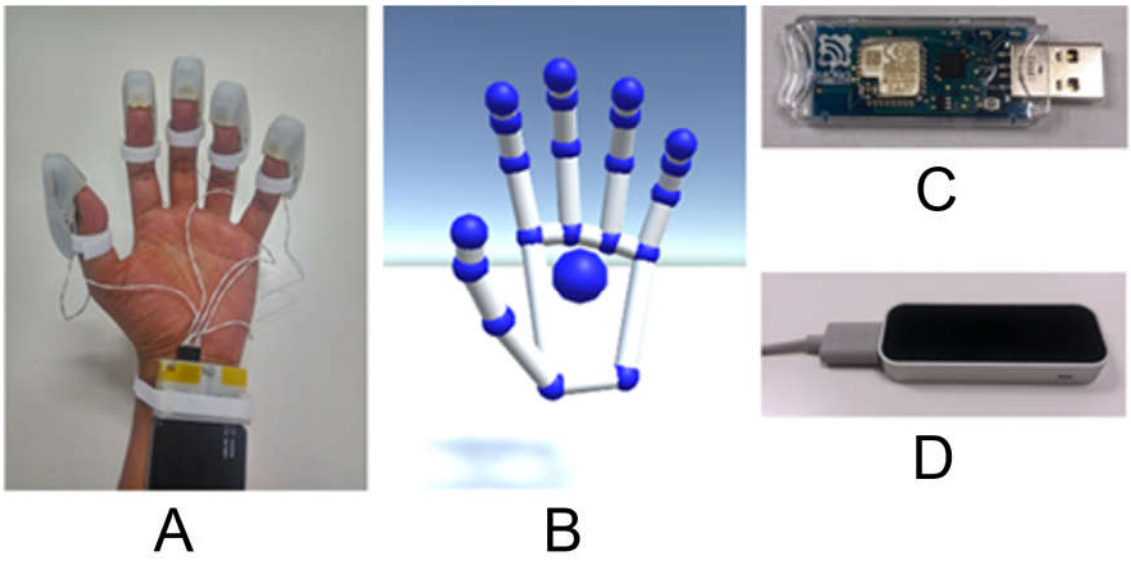


Figure 6: Haptic rendering device [10][11].
(A) Fingertips and wrist vibrating parts.
(B) Monostick USB interface.
(C) Leap Motion controller.

3.3 Haptic rendering

Three haptic parameters are provided by the haptic rendering device to render the tactile sense of virtual objects in a 3D simulation. When a real finger touches a real object, the first feeling is pressure from the surface of the object. Surface roughness is not quite noticeable unless the finger moves along the surface. Friction between the skin and the surface creates a small vibration that is getting bigger for a rougher surface. The faster the finger moves along the surface, the more frequent vibrations that can also be said that the frequency of vibrations is higher. We set the haptic parameters of the haptic rendering device according to this understanding. Pressure feedback is to simulate finger pressure / penetration to the surface of an object, the amplitude of the vibration is determined by the surface roughness of the object, and the frequency of the vibration is determined by the velocity of the finger moving along the surface.

3.3.1 Haptic rendering on simple objects

Before being used to render the haptic sense of molecular objects, the haptic rendering device was tested on ordinary virtual object. The Unity game engine was chosen as a framework for this implementation because it supported the required hardware. We used HTC

Vive head-mounted display (HMD) for the VR system and the Leap Motion controller for the hand gesture tracking, and they are well supported by Unity. Figure 7 illustrates the first implementation of the haptic rendering tool.

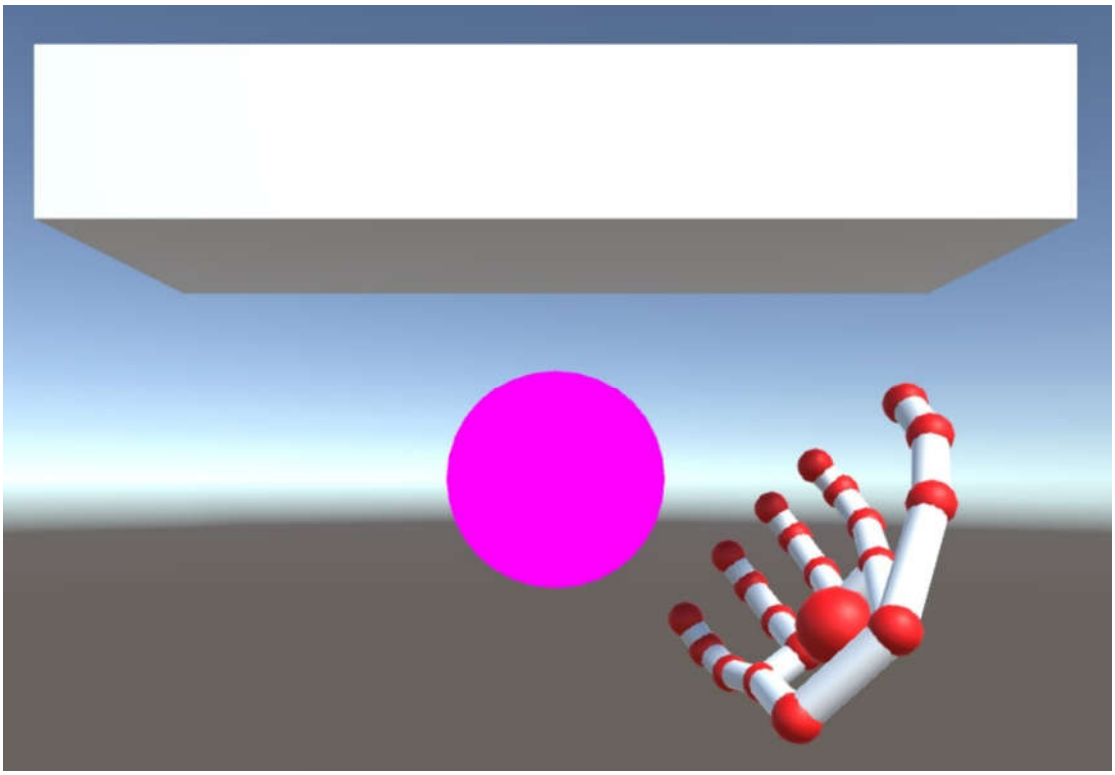


Figure 7: The first implementation of the haptic rendering device is at Unity. The purple ball and the white box are attached to their global position with springs that will always pull them back to their original position.

A purple ball and a white box were presented at the scene as floating objects to interact. Each object was held in its position relative to the simulation world (global position) with a spring. The spring will always try to maintain the position of the object by pulling it back to its original position every time it is moved by the user's hand. The haptic script was implemented in the OnTriggerEnter event of the finger which will produce a haptic response signal to be sent to the haptic rendering device every time the finger touches an object. When the finger leaves the object, an OnTriggerExit event occurs and stops generating a haptic response signal. For this implementation, each object was given the same haptic value which will produce the same vibration amplitude and pressure for tactile feedback. The vibration frequency was determined by the speed of the finger in moving along the surface of an object. The haptic device responded very well by always producing haptic feedback when touching any object.

3.3.2 Haptic rendering on heterogeneous surfaces

The second implementation was to render haptic feedback from several objects with different haptic values. Three boxes of different colors and different haptic values were arranged together into an object in the scene (Fig. 8A). The purpose of this arrangement was to test how well the haptic rendering device and the haptic algorithm currently provided tactile feedback from an object with a heterogeneous surface. The finger touching the object moved from one colored part to the other colored part. Haptic feedback was expected to change smoothly when touching different surface roughness. However, it did not produce feedback as expected. The feedback disappeared after moving to different colored parts of the object. This occurs due to the touch transition from one part to the neighboring part.

Figure 8B, 8B, and 8D illustrate a touch transition from one part (red part) to the adjacent part (green part). When the finger was moved to another part (Fig. 8C), the OnTriggerEnter event of the green part was triggered, even though it had previously been triggered by the red part. When the finger left the red part (Fig. 8D), the OnTriggerExit event of the red part was triggered, causing the haptic response signal to stop. In this case, the haptic feedback disappeared even though the finger was still touching the green part. To work around this problem, a counter variable was applied to count the number of OnTriggerEnter events minus the number of OnTriggerExit events that were currently occurring. The OnTriggerEnter event would increase the counter variable by one and OnTriggerExit would decrease it by one. With this approach, OnTriggerExit would know when the finger has released its touch with certainty. The haptic response signal would stop only if the counter variable became zero.

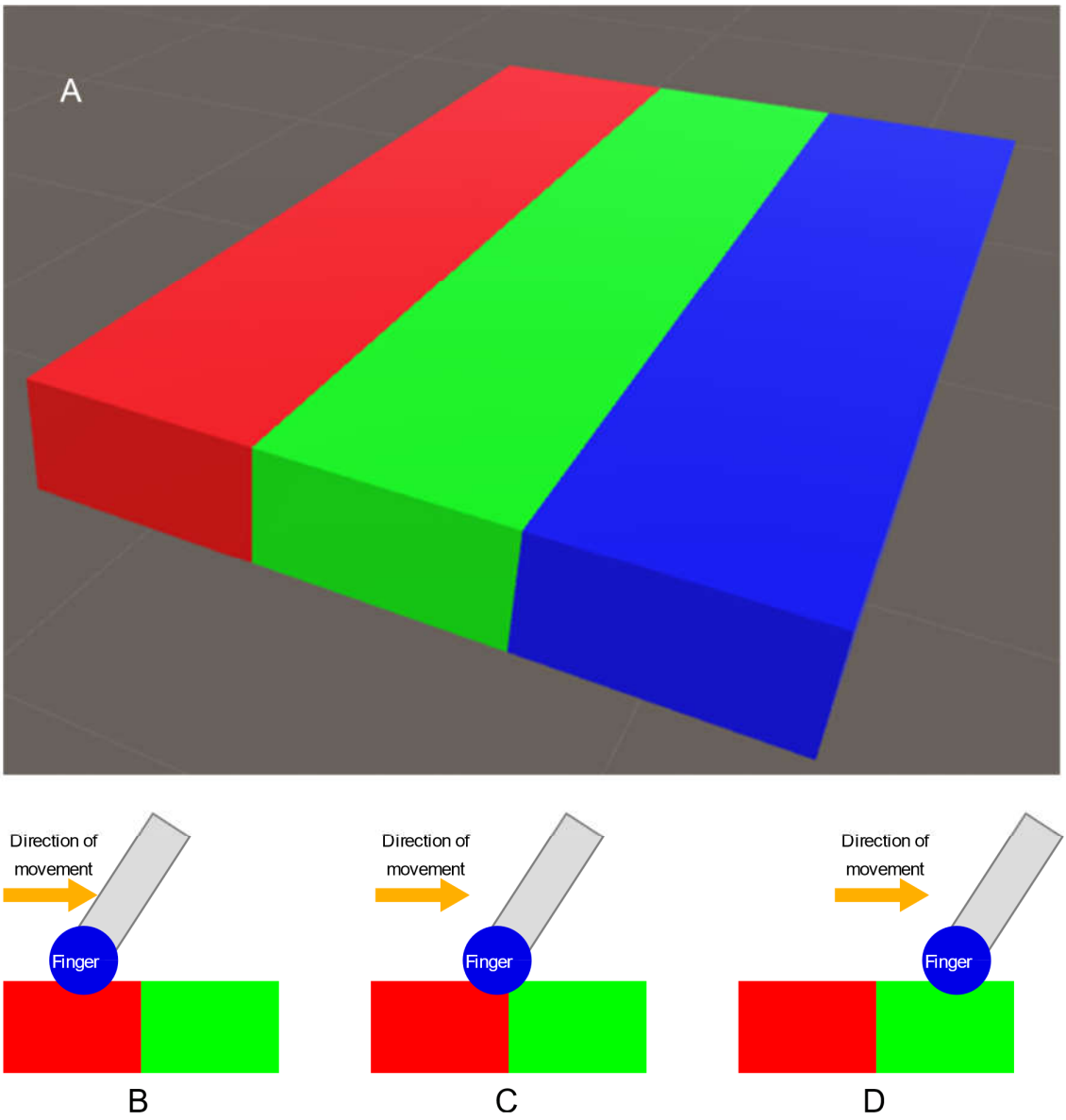


Figure 8: Renders haptic feedback from objects with different haptic values. (A) The arrangement of three boxes into an object with a heterogeneous surface. Each part with a different color has different haptic values. (B) The finger moves along the surface of this heterogeneous surface from one part (red part) to the adjacent part (green part). (C) The moving finger enters the green part before leaving the red part completely. (D) The moving finger comes out of the red part and is completely on the green part.

3.3.3 Haptic rendering on molecular objects

In principle, the haptic feedback generated when touching a molecular object in a VR simulation is similar to the haptic feedback generated when touching an ordinary 3D object. However, since it is impossible to touch a real molecular object, no one knows the feeling of

touching a molecule. Especially understanding molecular roughness remains a mystery. Therefore, to produce plausible haptic feedback, we must postulate the roughness of a molecular object using a value analogous to the vibration amplitude used to indicate roughness in the haptic feedback of an ordinary object. We propose to use the temperature factor values (b values) which can be found in each protein data bank file. The temperature factor applies to each atom indicating the uncertainty of the atom's position. This displacement of atoms can indicate the dynamic mobility of the atom, which higher value signifies lower confidence in its position or higher mobility. The temperature factor can be considered as an indication of how far the atom can fluctuate from its original position. Given the fact that each atom vibrates, the temperature factor can be equated as the vibration amplitude. Therefore, interpreting the temperature factor as the vibration amplitude of the haptic feedback makes sense.

In a coarse-grained simulation, objects are not represented as all atoms. Depending on the degree of coarse-grained, a group of atoms are represented as one grain (mesh object). To determine the vibration amplitude feedback, the temperature factor values of all the atoms in each grain are averaged and assigned to the grain. The implementations of each of the three haptic parameters for rendering touch to molecular 3D objects are:

- Pressure strength is determined by the finger pressure / penetration to the surface of the molecular 3D object.
- Vibration frequency is determined by the velocity of the finger moving along the surface of the molecule.
- Vibration amplitude is determined by the average temperature factor of all the atoms represented by the 3D mesh object being touched.

3.4 Molecular object creation

There are three types of representations that we use in this study: atomic model representation (Fig. 9A), ribbon model representation (Fig. 9B), and simple surface representation (Fig. 9C and Fig. 9D). Alpha beta tubulin dimer molecule from 1JFF.PDB file [61] was used in this and subsequent chapters to evaluate the haptic implementation and performance of the VR simulation system frameworks. The file contains property data for its 6704 atoms, of which the 3D position (x, y, and z coordinates) and the temperature factor are the most needed properties for this study. The atomic model representation requires only the 3D position of the atoms, while the other representations require preprocessing. UCSF Chimera molecular visualization [62] was used to translate PDB files into polygon mesh

objects, such as balls and sticks atomic models, ribbon models, and surface models. The selected polygon mesh object was then saved as an X3D format file that can be opened and manipulated using Blender [63] — a 3D computer graphics software. The polygon mesh object was edited and saved as other 3D file formats required for the simulation, such as 3DS and FBX files.

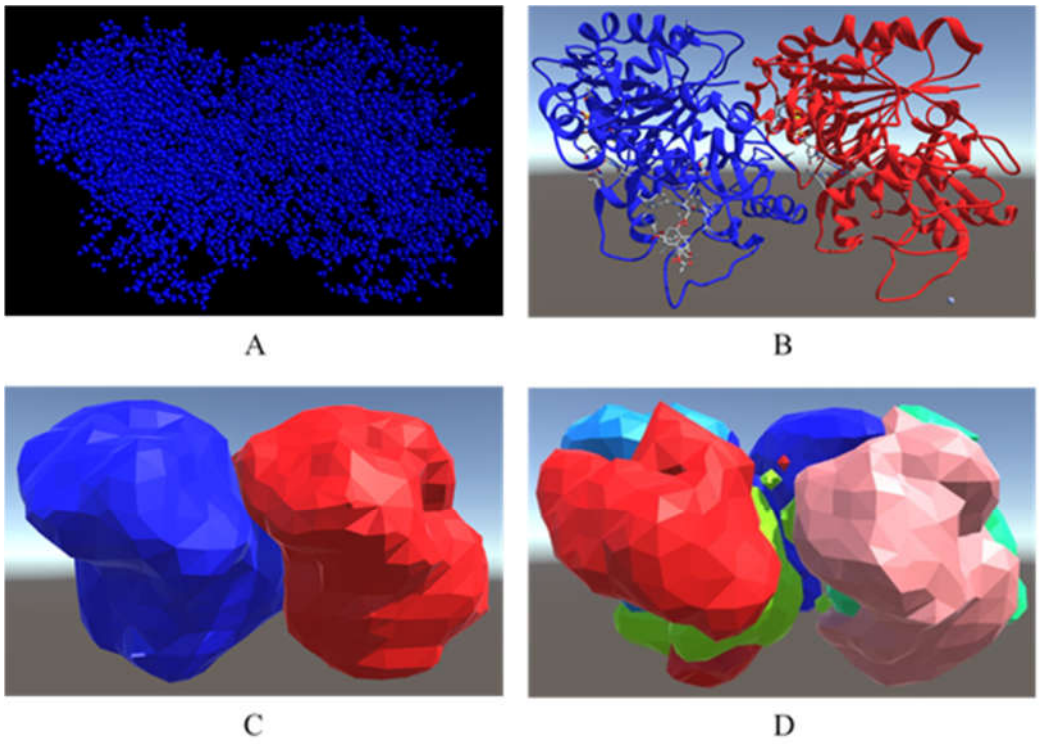


Figure 9: Three types of representation in this study using alpha beta tubulin dimer molecule from the 1JFF.PDB file.
(A) Atomic model representation.
(B) Ribbon model representation.
(C) Simple surface model representation.
(D) Simple surface model representation with segmentation.

Coarse-grained models using surface models were used to create simple surface models (Fig. 9D). The atoms of alpha beta tubulin dimer were segmented by the k-means clustering method, featuring 3D coordinates and temperature factor to group the atoms into six groups. To determine the haptic vibration amplitude feedback for each group, the average temperature factor values of the atoms in each group were calculated. Six polygon mesh objects were generated by UCSF Chimera from this group of atoms and then combined into one file using Blender. This coarse-grained model manages to distinguish temperature factors for haptic vibration feedback in each part of the dimer.

3.5 Haptic rendering implementation in different simulation frameworks

Our haptic rendering concept for touching molecular 3D objects is proposed for the ideal haptic device and simulation system which fully supports three haptic parameters. In their implementations, haptic devices and simulation systems may support haptic feedback in a non-ideal way. Therefore, the implementation of the haptic rendering concept is slightly different for each simulation framework. In this study, we apply the haptic rendering concept for touching molecular 3D objects in three simulation frameworks: CHAI3D, Unity, and particle simulation with DirectX 12.

3.5.1 Haptic study of touching molecules on CHAI3D

We studied how haptic parameters should determine haptic feedback in a molecular modeling environment using the CHAI3D framework (version 3.2) for the Geomagic Touch haptic device. The framework and the device provide various examples of impressive haptic feedback which were of great help in development. This grounded device can perform complex haptic feedback including force feedback, which normally cannot be done with ungrounded devices. Therefore, the results can be implemented for our haptic device to achieve a good haptic impression. Figure 10 shows the CHAI3D molecular viewer prototype with the Geomagic Touch haptic device.

The CHAI3D program works by running a graphic thread and a haptic thread simultaneously with a haptic thread that has priority. As a non-VR graphics program, the ideal graphical framerate is 60 Hz (60 frames per second). Meanwhile, the haptic thread is required to run at 1000 Hz (1000 updates per second) for Geomagic Touch devices to provide responsive haptic feedback. Consequently, graphics performance is sometimes sacrificed by haptic performance needs.

Reading molecular data from PDB files was implemented in C++ to visualize all atoms as spheres on the scene and attach temperature factor values for each atom. Users can interact with atoms using the Geomagic Touch device. The device provides a spherical cursor called a haptic cursor to explore and touch atoms. This program worked well for molecules with a small number of atoms. However, running it for a molecule with vast number of atoms resulted in a significant reduction in performance. This might be due to the default sphere rendering parameter with gluSphere() in OpenGL.

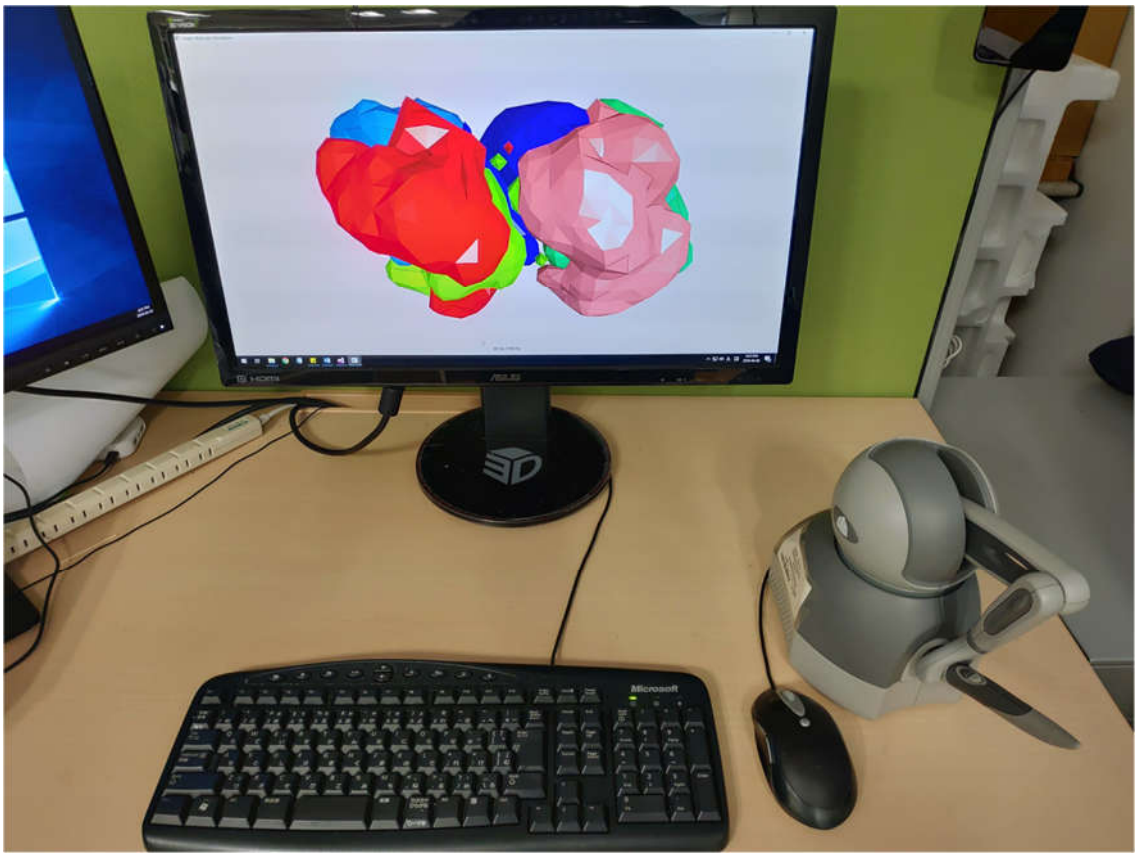


Figure 10: CHAI3D molecular viewer prototype with Geomagic Touch haptic device.

CHAI3D can be categorized as a graphics framework with strong support in haptics but does not yet have physics simulation. By default, CHAI3D does not use a physics engine for object interactions. It provides haptic interactions between the haptic cursor and the 3D objects, but the objects themselves do not collide with each other. In fact, the objects only move if the user moves it with the haptic cursor. Therefore, objects in CHAI3D can be considered as static objects.

Vibration feedback for the haptic was determined by our code, while the pressure feedback was determined by the CHAI3D default algorithm for supported Geomagic Touch device. It used potential field and finger-proxy models [64] to prevent the haptic cursor from penetrating any object while also generating force feedback to push the user's hand back. Vibration frequency feedback was determined by the speed of movement of the haptic cursor along the surface of the molecular 3D object, whereas amplitude feedback was determined by the temperature factor of the molecular object touched.

Haptic rendering with Geomagic Touch successfully provided tactile feedback that was able to distinguish parts of objects that have different amplitude parameters. The vibration

feedback implementation ran according to the scenario so that the user running the haptic cursor along the surface of the object would feel a tactile sensation as if touching the surface of an object with different roughness. The ability of Geomagic Touch to prevent penetrating objects touched by the haptic cursor was in accordance with the static object properties found in CHAI3D. This showed that our haptic rendering concept was working properly.

3.5.2 Haptic molecular viewer on Unity

The ten-finger haptic rendering device requires a Leap Motion controller to deploy in a VR environment. Unity, a well-known game development software platform, was used with the advantage of being widely supported by major vendors including Leap Motion controller. It supports HTC Vive HDM device, enabling the development of a VR molecule viewer in room-scale VR. Unity also provides a physics engine PhysX for rigid-body object interactions. The user's hands were tracked by the Leap Motion controller and projected in the VR scene to interact with the VR object.

Ribbon model representation of molecules was used to present large molecular objects in a fine detail. Figure 11 shows a ribbon model representation and the hand model in Unity. The object can be grasped and moved by hand. The number of polygon mesh objects in this representation was not as big as the atomic model, but it still showed good detailed atomic information. We experienced that PhysX ran smoothly when the number of objects in one scene was less than a few hundred. However, the performance drops dramatically when the number of objects reaches thousands. Touch feedback was calculated for update events, so it was calculated for each graph frame update. This caused haptic feedback to be synchronized with events that occurred at the scene. The user would get the same tactile feeling he saw on the screen. Therefore, the user would not experience any difference between what he saw in the scene and what he felt when touching, regardless of the performance framerate.

The vibration frequency was adjusted to the movement velocity of the fingertip along the surface of the molecular object being touched. It was calculated by the distance between the position of the fingertip in the current frame and its position in the previous frame divided by the time difference between those frames. The vibration amplitude was determined by the average temperature factor of all the atoms in the polygon mesh object that were in contact. Ideally, pressure feedback should be determined by penetration of the finger through an object. However, the rigid body physical calculation would most likely make the object pushed away by the hand in every contact. Hence, finger penetration was unlikely to happen. To make use

of the pressure feedback, it was set linearly with the amplitude value to enhance the tactile sense. As a result, this clarifies the difference in amplitude values.

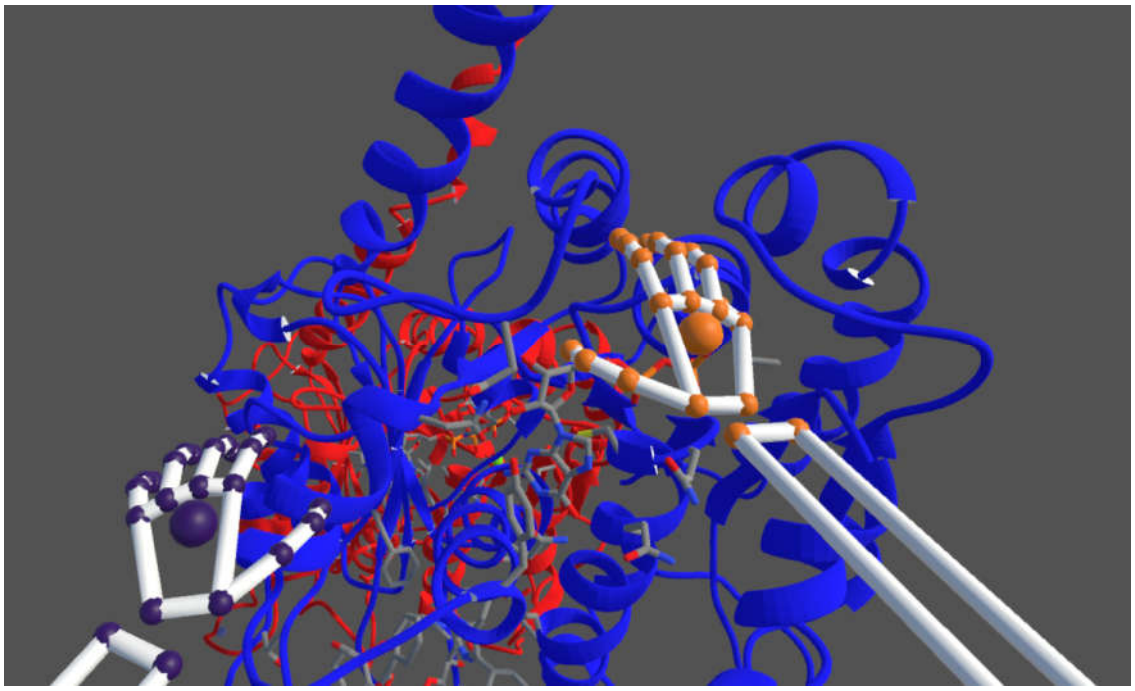


Figure 11: Atomic model representation of the 1JFF alpha-beta tubulin dimer in Unity with the hand models from the Leap Motion controller.

3.5.3 Haptic rendering in molecular haptic particle simulation with DirectX 12

DirectX 12 is a Graphics API that allows compute shaders to utilize of the GPU for simulated compute. Being a low-level API, DirectX 12 allows deep optimization for optimal simulation performance. GPU computing even greatly improves performance. However, everything must be built from the ground up. The OpenVR SDK was implemented so that the HTC Vive HMD device for VR can be enabled. The Leap Motion controller was activated using its LeapC SDK. The haptic rendering device code driver and its haptic algorithm for haptic rendering must be developed with our own C++ coding.

This molecular haptic particle simulation system was developed based on previous work in the microtubule gliding assay simulation [12][13][14]. Every object in the simulation was created from particles — spherical entities with position, velocity, and radius. Particles were connected by springs to form compound objects. The collisions between particles were

calculated using the Lennard-Jones (L–J) potential formula. For certain conditions, the L-J potential was modified by eliminating the attractive force between the particles, leaving only the repulsive force as a simple collision interaction.

Since all objects are made of particles, the best molecular representation for this system is the atomic model representation because atoms are described as spherical bodies. In a more coarse-grained level, a particle can represent a group of atoms occupying the space of the particle. The number of atoms represented by a particle is highly dependent on the level of coarse-grained of the simulation. The particles connect the nearby particles within a certain distance to form the shape and structure of the object. This network of springs allows particles to move or deform as a group when another object collides with them, and then they will always try to return to their original shape. For example, when a virtual hand grasps or pushes the object.

The virtual hand is also formed by particles based on Leap Motions tracking data, which has a tracking frequency of 115Hz. This gives the physical properties of the hand in the simulation and allows to monitor the collision force of the particles to the hand to create very precise haptic feedback for the user. The haptic feedback is calculated by the number of interaction forces applied to the hand particles with the object's particles. All hand particles representing each fingertip and wrist are identified and grouped. The collision forces are accumulated from all the hand particles in each group. The vibration and pressure feedback of each fingertip and the wrist will be determined by these accumulated collision forces. For reasons of simplicity, haptic feedback is currently only divided into five levels of value with the amplitude and pressure adjusted linearly to differentiate the tactile sense.

An alpha beta tubulin dimer as 3D objects in this simulation is shown in Figure 12. The atoms in this molecule were pushed and the shape of the molecule is deformed by the virtual hands. After the pressure from the virtual hand was removed, these atoms arranged themselves into their original molecular form with the object possibly having changed its position and orientation. This shows visually that molecular objects are soft and deformable. However, it is still difficult to understand how soft these molecular objects are. The haptic rendering that we provide can help users to understand the forces in the simulation more easily after interacting with them. This makes the simulation more immersive and intuitive.

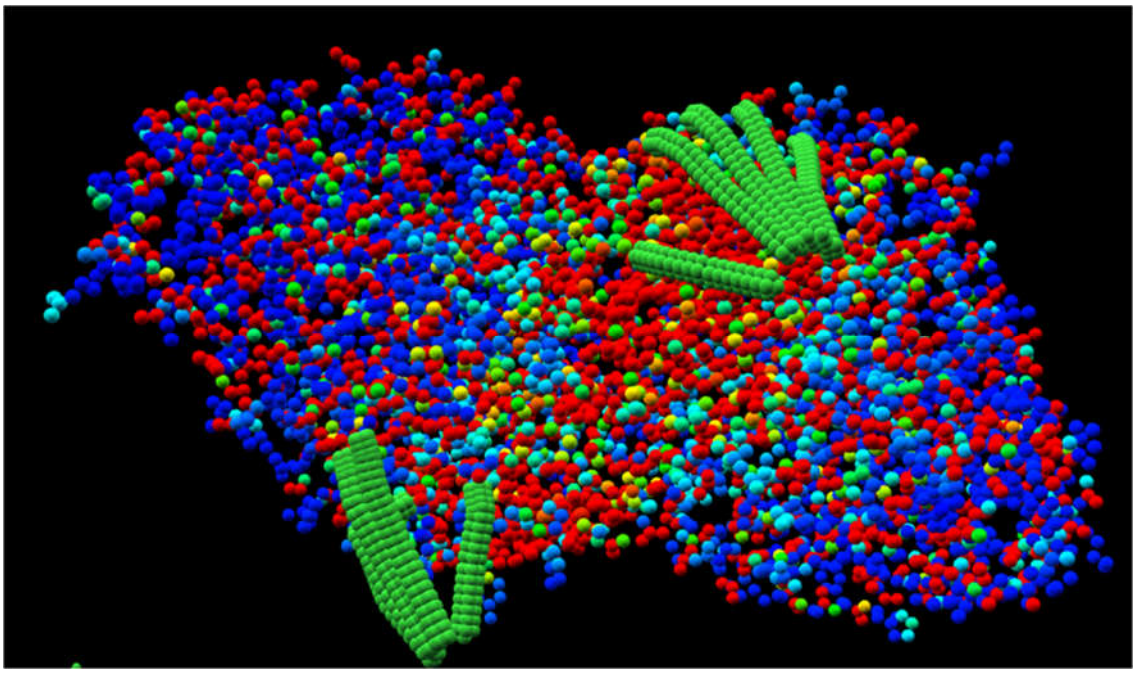


Figure 12: Atomic model of alpha beta tubulin dimer in DirectX12 particle simulation. Two virtual hands are made of particles moving in groups according to hand tracking data captured by the Leap Motion controller.

3.6 Comparison with other devices

The development of haptic technology and VR simulation has led to a wide selection of supporting devices. The simulation platform that we developed in this study uses devices that also have other alternatives. Several comparisons between related devices are shown to understand the advantages and limitations of hardware for hand user interface with the haptic feedback it provides.

3.6.1 Hand tracking controller comparison

As previously mentioned, the Leap Motion controller is the hand tracking controller we used in for our simulation platform. It is an optical tracker designed to continuously track full hand movements, especially palms and fingers. It commercially targets general consumers and gamers; hence, it is affordable and easy to use. The vendor has provided good support for the development of their devices by providing the Leap Motion SDK which is integrated with Unity and the Unreal engine. The SDK is provided in various programming languages such as C ++, Objective-C, C #, Java, and Python to support its development.

A similar device which is also an optical motion tracker is the Microsoft Kinect. This device

can track the entire user's body. It uses a 640×480 pixels RGB camera at 30Hz framerate and a 640×480 pixels infrared depth-finding camera also at 30Hz framerate. It has an operating range between 0.5 m to 5 m according to specifications [32]. The vendors support the development very well by also using the device in their famous Xbox 360 game console.

The differences in capabilities between the Leap Motion controller and the Microsoft Kinect controller are:

- The Leap Motion controller average accuracy is 0.7 mm, while the Microsoft Kinect controller accuracy ranges from a few millimeters to about 4 cm.
- The maximum effective operating range of the Leap Motion controller is up to about 1 meter, while the Microsoft Kinect controller has a maximum effective operating range of 5 meters.
- The Leap Motion controller can generate up to 200 frames per second of data according to its documentation [60] or 115 frames per second of data in our experience, whereas the Microsoft Kinect controller produces 30 frames per second of data.

Based on these data, it appears that the Leap Motion controller offers a smaller observation area, but a higher resolution compared to the Microsoft Kinect controller. The Leap Motion controller is designed for full hand tracking, while the Microsoft Kinect controller is designed for full body tracking. For the purposes of user interaction with molecular objects in the VR environment that we have developed, hand tracking accuracy and precision are more important than full-body tracking capabilities. Because of this, the Leap Motion controller is superior to the Microsoft Kinect controller in our purpose.

3.6.2 Comparison with grounded haptic devices

Grounded haptic devices such as Geomagic Touch have become the preferred choice for haptic feedback interfaces in most molecular modeling. Being attached to a strong fondation gives them the ability to generate motion force feedback to create more complex and realistic 3D object feedback without penetrating the surface. However, because molecular objects are described as very soft and deformable objects, this benefit is not important as penetration of the object with virtual hands is permissible. The additional benefits of these device may also be disproportionate to the high cost of the devices for some researchers. Size, weight, and mechanical complexity also limit their mobility; therefore, they are not suitable for a room-scale VR environment.

On the other hand, the noticeable component of the haptic rendering device is the Leap

Motion controller, which is inexpensive. Due to their low cost, portability and usability, Leap Motion controllers have been widely used in research as well as in the gaming industry. The compactness and mobility of this device make it suitable for any VR environment, including room-scale VR environments.

In terms of interaction points, the available grounded haptic devices use a single spherical haptic point or haptic cursor to interact with 3D objects and obtain haptic feedback. Apart from being used for pushing, the cursor can be used to hold, move, and rotate an object by implementing programming code that utilize the device button. When the haptic cursor touches an object, pressing the device button can make the object attach to the cursor to allow it to be moved and rotated. However, this interaction method only works to handle one object at a time.

Our simulation platform offers a natural user interface by providing full hand interaction with 3D objects. Virtual hands provide a comfortable, natural way to interact with molecular objects. Holding two or more objects at a time is almost as natural as in the real world. The area stretched by the palms and fingers can be used to push molecular objects from a specific area to deform the object. It is even possible to scoop up some small objects with virtual hands. The ten fingers and two wrists haptic points of the haptic rendering device used on our platform provide an intuitive advantage over grounded haptic devices.

3.6.3 Comparison with other hand haptic devices

The haptic rendering device used in our simulation platform is a kind of hand haptic device with vibration feedback. Since the number of hand haptic device is large, it is best to narrow the comparison with hand haptic kits that have vibration feedback. We identified three similar devices in this category: Grabity, Go Touch VR, and Manus VR Prime II Haptic. The comparison can be seen in Table 2.

Go Touch VR and Manus VR Prime II Haptic are the closest to the haptic rendering device we are using. Both provide a fully wireless connection and full hand interaction with vibration feedback at the fingertips just as the haptic rendering device. They also simulate object softness but lack a mechanism to prevent virtual hands from penetrating objects. The difference with the haptic rendering device is that Go Touch VR only provides feedback for three fingers per hand and Manus VR Prime II Haptic has no finger pressure feedback. In terms of the number of haptic points, the haptic rendering device has more haptic points in addition to the two wrists. Overall, the haptic rendering device we are using has the advantage of more haptic points and included a finger pressure feedback feature.

Table 2: Features comparison between the haptic rendering device and other hand haptic devices.

Features	Haptic rendering device	Grabity	Go Touch VR	Manus VR Prime II Haptic
Vibration feedback	**Yes**	**Yes**	**Yes**	**Yes**
Force feedback	**Finger pressure**	**Finger pressure**	**Finger pressure**	No force feedback
Penetrating objects	Does not prevent object penetration	**Prevent object penetration**	Does not prevent object penetration	Does not prevent object penetration
Number of haptic points	**Five fingers & one wrist per hand, two full hands**	Two finger, one hand	Three finger per hand, two full hands	Five fingers per hand, two full hands
Freedom of movement	**Fully wireless**	Limited by a connection wire	**Fully wireless**	**Fully wireless**
Simulates softness	**Yes**	Does not simulate softness	**Yes**	**Yes**
Simulates object's weight	No	**Yes**	No	No

Grabity is the most different compared to the others previously mentioned. This device only provides one-handed interaction with two haptic points. It requires a wired connection to the host computer. Although it has a unidirectional breaking mechanism to create object's stiffness when grasped by the hand, it does not simulate object's softness. This breaking mechanism gives the device the ability to prevent the hand from penetrating objects, something other devices cannot. However, as we mentioned earlier that molecular objects are soft and deformable, objects penetrated by hand are permissible. Therefore, this advantage is neglected.

Another feature that only Grabity has is that it simulates the weight of an object. By generating asymmetrical vibrations, Grabity generates a virtual force that reproduces the sensation of the weight of an object for the user when holding it. This is interesting feedback for simulating real-world objects; however, gravity is usually neglected in the molecular world. Therefore, this advantage is also not very beneficial. Comparison summary of the haptic

rendering device and the other hand haptic devices can be seen in table 3.

Table 3: Comparison summary of hand haptic devices.
Good marks are given to devices that benefit from the statement mentioned in the left column. Bad marks are given for the opposite.

	Haptic rendering device	Grabity	Go Touch VR	Manus VR Prime II Haptic
Molecular objects are very soft and deformable; therefore, penetrating object is permissible.	**Good**	Bad	**Good**	**Good**
Useful parameters: vibration frequency, vibration amplitude, and pressure.	**Good**	Bad	**Good**	Bad
Gravity is not significant.	**Good**	Bad	**Good**	**Good**
Fully wireless device provides freedom of movement.	**Good**	Bad	**Good**	**Good**
Full hand and finger haptic feedback provides intuitive interactions.	**Good**	Bad	Bad	**Good**

3.7 Summary of the natural user interface

The natural user interface we provide for the VR simulation platform consists of a hand tracking controller and a haptic rendering device. We chose to use the Leap Motion controller for the hand tracking controller over the Microsoft Kinect controller as it has better accuracy and precision. The Leap Motion controller has an average accuracy of 0.7 mm which is better compared to the Microsoft Kinect controller accuracy which ranges from a few millimeters to about 4 cm. For streaming data, the Leap Motion controller leads up to 200 Hz compared to only 30 Hz in the Microsoft Kinect controller. For a full-handed VR interface without showing the body, the Leap Motion controller is better than the Microsoft Kinect controller.

Hand haptic devices as ungrounded haptic devices are better for VR compared to grounded haptic devices. The size, weight, and mechanical complexity of grounded haptic devices make them inconvenient to use in VR, especially in room-scale VR environments. Hand haptic devices also have a higher number of interaction points compared to just one haptic cursor that a grounded haptic device has. With a hand-like user interface, hand haptic devices provide a

natural user interface for more intuitive user interaction.

Compared to other hand haptic devices, the haptic rendering device we used in this study was better for molecular VR simulations. It has more haptic points than others mentioned in this study. It has a finger pressure feedback feature that the Manus VR Prime II Haptic does not have. Fully wireless connection to the host computer is an advantage for freedom of movement in a room-scale VR environment whereas the Grabity haptic device has limited mobility due to its wired connection to the host computer. Haptic rendering devices do have limitations in field of view, accuracy, servo motor components, and hand interaction force. However, they can still be handled wisely so that the VR simulation runs smoothly.

The contributions we offer in the development of natural user interfaces for our VR simulation platform are:

- It provides a natural hand user interface with haptic feedback in molecular VR simulations for more intuitive interactions.
- We propose a concept on how to render tactile feedback when touching a molecular object in molecular simulations using three haptic parameters: vibration frequency, vibration amplitude, and pressure strength. Notably the idea of using temperature factor values to interpret the surface roughness of the molecules.

3.8 Molecular VR simulation with existing simulation framework

Two important aspects of 3D simulation are graphics performance and computation performance. A simulation framework consists of at least a graphics API and a simulation engine to perform simulation calculations. To fulfill the requirements of molecular VR simulation, the simulation framework must be able to provide a graphics performance of at least 90 Hz while rendering the views of the left and right eye. In addition, to simulate molecular phenomena, the framework must be able to simulate many objects in that performance. Although there are many simulation platforms, this study takes three simulation platforms that have the potential to be used in the development of an interactive haptic VR simulation platform for molecular phenomena: CHAI3D, Unity, and DirectX 12 particle simulation system.

The alpha beta tubulin dimer molecule that we used in this research was presented in three models: simple surface model, band model and atomic sphere model. Each model requires a different geometric complexity because more complex mesh objects require more geometric vertices and triangles (faces) to create. It should also be noted that although the atomic sphere

model uses a simple spherical object, the number of objects required is very large because each atom must be represented as a sphere. Therefore, the amount of geometric data is enormous. The geometric data for each model representation can be seen in Table 4.

An attempt to reduce the geometric data on the atomic sphere model by reducing the number of triangles per sphere helps improve performance. An atomic sphere model with 120 triangles requires 29 vertices per sphere, while 80 triangles require 46 vertices, and 48 triangles require 29 vertices. However, how significant the performance increase should be considered because it affects the object's appearance. Figure 13 shows the different appearance of spheres with different resolutions.

Table 4: Geometric data of 1JFF alpha-beta tubulin dimer representations.

Model	Number of vertices	Number of triangles	Number of mesh objects
Simple surface	2300	4516	6
Ribbon	233668	348432	1692
Atomic sphere model with 48 faces	194416	321792	6704
Atomic sphere model with 80 faces	308384	536320	6704
Atomic sphere model with 120 faces	449168	804480	6704

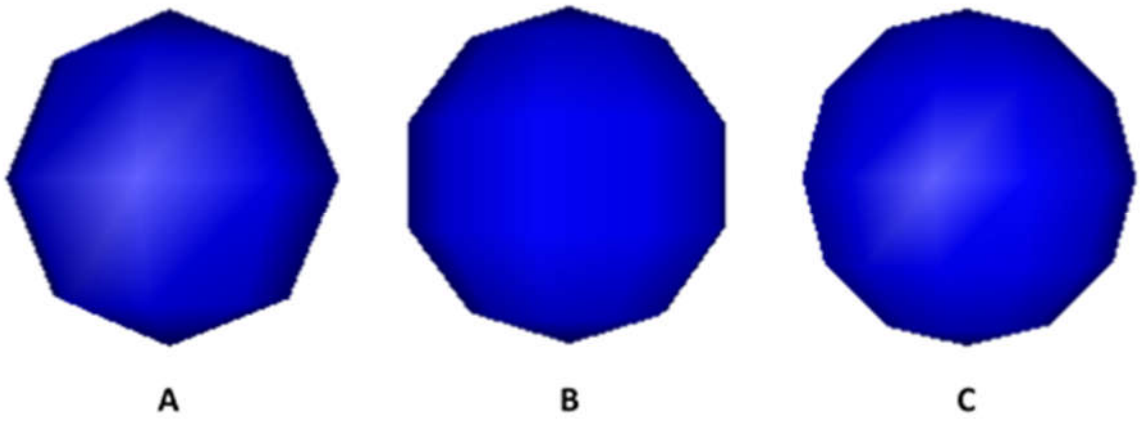


Figure 13: The appearance of spheres in atomic sphere models with different resolutions. (A) With 48 triangles. (B) With 80 triangles. (C) With 120 triangles.

3.8.1 Molecular haptic simulation in CHAI3D

CHAI3D is a simulation framework specialized in haptic simulation that supports a wide variety of haptic devices. It has been widely used to develop haptic systems for various purposes. The graphics API is based on OpenGL. It does not have a built-in physics engine, although as an open-source framework makes it possible to implement its own engine by coding from the ground up. What it does provide is a haptic simulation package that simulates haptic feedback using any supported grounded haptic device. Collisions between the haptic cursor and objects are provided only to generate haptic feedback for the cursor. Therefore, the collision will not move the object unless an additional algorithm is applied.

The haptic simulation is executed on a dedicated thread that separates it from the graphical process. Since it is optimized for haptic simulation, haptic threads have priority over others by default. Because of this, graphics performance usually suffers the most when it comes to heavy computational work. In addition, the ideal haptic rate for common grounded haptic devices is around 1000 Hz, exploiting most of the computing resources.

Molecular objects in CHAI3D were represented in all three models. When represented in a simple surface model, the simulation ran at the expected 60Hz graphical framerate in dual display setting. This was the maximum framerate for the CHAI3D as it was not aimed at VR performance. Maximum haptic speed was also performed at usual 1000 Hz. This was understandable because the number of objects was only six without geometric complexity.

The ribbon model representation also ran well in CHAI3D framework. Although the amount of geometric data has increased from thousands to hundreds of thousands and the number of mesh objects has increased from six to 1692 compared to the simple surface model (Table 4), the graphical framerate has reached 58 Hz, which is not too far from the maximum of 60 Hz.

A significant decrease in performance occurred in the atomic sphere model representation. The amount of geometric data in this model overwhelmed the simulation system, which resulted in reduced graphics and haptic performance. The worst performance occurred when the haptic cursor touched and moved along the object surface (Table 5). Although the number of vertices and triangles of the low-resolution atomic sphere model (with 48 triangles per sphere) was lower than that of the ribbon model, the performance was much lower due to the higher number of mesh objects. This was a sign that the haptic simulation calculations are bottlenecks in this performance decline. More collision objects in the scene resulted in lower performance. These results proved that the reduction in geometric resolution for the atomic

sphere model representation plane in the CHAI3D framework does not significantly help performance.

Table 5: Performance of the CHAI3D framework for 1JFF alpha-beta tubulin dimer in atomic sphere model representations of different geometric resolutions.
Taken when the haptic cursor interacts with the model.

Model	Graphics framerate (Hz)	Haptic rate (Hz)
Atomic sphere model with 48 faces	12	250
Atomic sphere model with 80 faces	9	240
Atomic sphere model with 120 faces	7	190

3.8.2 Molecular haptic simulation in Unity

We implemented the haptic rendering device and the Leap Motion controller in the Unity framework. As a well-known gaming software development framework, Unity is widely supported by major hardware vendors including Leap Motion and HTC Vive vendors. Room-scale VR environments are provided using the HTC Vive HDM device. Support by major hardware is helpful for fast and easy VR simulation system development.

Molecular objects in Unity were also represented in all three models. However, due to performance issues, the atomic model in this implementation was not an ideal object implementation as each atom was only constrained by a string to a global position at the scene. This implementation was for performance comparison purposes only. All model representations were implemented in VR simulation with PhysX as the active default physics engine. All GPU resources were used for graphics work because PhysX did not use the GPU for computing.

The graphical framerate of a simple surface model representation reached 140 ~ 160 Hz, which was great for VR. However, in the ribbon model representation, the performance dropped to 25 ~ 30 Hz. For VR simulations, this performance was not satisfactory, even though it was still acceptable for short time usage. When the number of objects increased, performance decreased even lower. Therefore, the atomic sphere model representation was below 20 Hz.

We also noticed the fluctuation in performance whenever a movement occurred, whether it was from a virtual hand or from another object. For further analysis, physics simulation testing was carried out on the atomic sphere model representation with three setups: without

movement, movement of virtual hands only, and movement of all atoms. For the third setup, a large and heavy spherical object was simulated to hit all atoms at once to move all atoms. Furthermore, each of these conditions was carried out on the model with all atoms (6704 atoms), with reduce number of atoms to 2000, and with 1000 atoms. Table 6 shows the results of this test.

Table 6: Graphical performance of the Unity framework for atomic sphere model representations in different setups and different number of atoms.

	Graphical Framerate (Hz)		
	1000 atoms	2000 atoms	6704 atoms
Without movement	108	49	16
Virtual hands movements	60	33	5.5
All atoms movements	22	1.1	0.2

According to the results in Table 6, both the number of objects and the number of movements at the scene significantly affect system performance. The two parameters represent the physics calculations. Therefore, it can be concluded that the physics engine was the bottleneck in this performance. Only relying on CPU computing, PhysX was overwhelmed by the burden of physics calculations. Therefore, the use of GPU for physical calculations seems to be important for simulating large objects.

3.8.3 Molecular haptic simulation in DirectX 12 particle simulation system

The particle simulation system generates a large-scale of particles in the simulation. The physics calculations are about simple particle physics but with massive numbers. Its calculations can easily be divided per particle physics. Therefore, this kind of system is very concurrent and parallel, which is very suitable for processing with parallel multiprocessors such as GPUs. As a matter of simple parallelism, the speed up can be so high that the increase in the maximum number of particles that can be simulated is almost linear as the number of processing cores increases [12].

The DirectX 12 particle simulation system in this study uses a simple and lightweight Euler integration in computing. All forces affecting particle physics are calculated per computational iteration (computation cycle). Each force will contribute to the movement of the particles by

increasing their velocity in a certain direction. The sum of all the forces in each particle per computation cycle determines the motion of the particles for the next computation cycle. This Eulerian process is described in Figure 14.

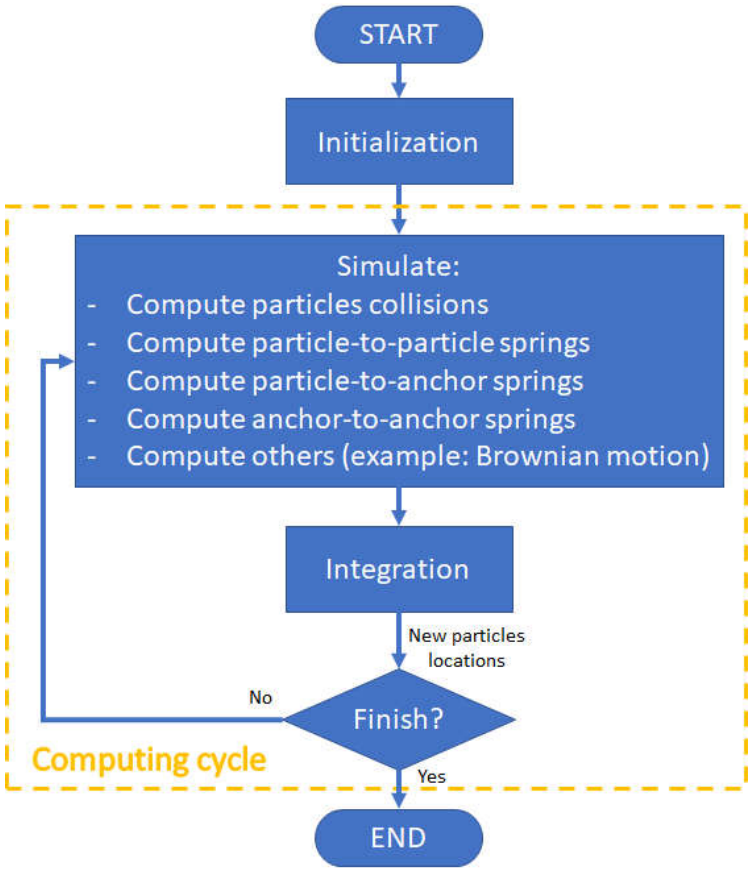


Figure 14: Flow chart of the DirectX 12 particle simulation system with Euler integration. In each computation cycle, all the forces in each particle are calculated as the additional velocity to move the particle for the next computation cycle.

The molecular objects in this particle simulation system were only represented in the atomic sphere model. The shape of the spherical particles is identical to the atoms in the molecule. With the radius setting to match the radius of each atom, the particle is the best representation of the atom. The shape of the ribbon model is difficult to represent by particles. If forced to represent the ribbon model, the required number of particles will be closer to the number of particles in the atomic model. Therefore, there is no point in using particles for the ribbon model. A simple surface model is possible to represent the particles covered by a deformable mesh surface. However, applying a deformable net surface is more complicated. Therefore, we will save it for future projects.

The particle simulation system worked well when representing atomic sphere models.

Performance over 120 Hz was always shown in atomic sphere model simulations with 48, 80, and 120 triangles per sphere. Since the particle simulation system performs simple parallelism, the calculation load in each computing cycle will be the same. Therefore, there is no fluctuation in performance at different collision intensities. For the record, GPU computing was only used for collision calculations when performance was being measured. Running all calculations and all data on the GPU will make a much more significant performance increase as data transfer between CPU memory and GPU memory is minimized.

One of the best performances of the DirectX 12 particle simulation system was the use of the NVIDIA GPU TitanX Pascal. Simulations of more than 100,000 particles with a maximum of 200 springs per particle were carried out at 90 Hz graphical framerate. The numbers can go up or down according to different simulation implementations. Moreover, the use of more than one GPU can certainly increase performance. To implement more than one GPU in a simulation is beyond the scope of this study. Another advantage of using DirectX 12 is its low-level graphics API, which allows for in-depth optimization by not using unnecessary graphics features that overwhelm computation work.

3.9 Summary of the VR simulation system

The performance of a molecular simulation depends on the objects used and the simulation framework used. Each object model generates a different amount of geometric data which will affect the computational load. Compared to graphics load, physics / haptic simulation load has more effect on system performance. This is related to the framework used. Frameworks that process simulations using the CPU will be overwhelmed by processing large number of simulated object calculations, while frameworks that use parallel processing from the GPU will easily process them. A summary of the performance comparisons of the framework is shown in Table 7.

Most of the existing VR molecular simulations are developed using Unity, where the PhysX physics engine only uses the CPU to compute the simulation. As a result, the number of objects supported by the existing system is not too large. Since simulations of natural molecular phenomena often require very large number of objects, these existing systems are unlikely to be the right choice. On the other hand, the DirectX 12 particle simulation system we used in this study efficiently parallelizes the computational load and optimally uses the GPU for simulation computations. This allows large number of objects to be simulated on the scene; therefore, it could potentially be used as a simulation engine for the interactive haptic VR

molecular simulation platform that we developed in this study.

Table 7: Graphical performance of the three simulation frameworks for molecular simulations with various object model representations.
In this comparison, a stereo dual display mode was used on the CHAI3D, whereas the Unity and DirectX 12 particle simulation systems use VR displays.

Model	Graphical Framerate (Hz)		
	CHAI3D	Unity	DirectX 12
Simple surface for volume	60	140	-
Ribbon model for backbone	58	30	-
Atomic sphere model with 48 triangles	12	0.2 - 16	> 120
Atomic sphere model with 80 triangles	9	0.2 – 16	> 120
Atomic sphere model with 120 triangles	7	0.2 - 16	> 120

3.10 Achievements

In this chapter, we have established a VR simulation engine and a hand user interface for the interactive haptic VR simulation platform. The achievements of this chapter are as follows:

- **Hand user interface with haptic feedback for the VR molecular system has been implemented.**

  The hand user interface allows the user to interact naturally with molecular 3D objects. Haptic feedback provides additional insight into observing interactions and intuitively sensing object dynamics. The hand user interface with the haptic feedback feature emphasizes the meaning of interactive simulation and makes parameter optimization more intuitive. All existing VR molecular simulations mentioned in this thesis do not provide this feature.

- **The haptic rendering concept for touching molecular objects has been proposed.**

  The haptic rendering concept interprets the tactile sense of molecules into three haptic parameters: vibration frequency, vibration amplitude, and pressure strength. The vibration frequency is used to represent the finger moving along the surface of the molecule, the vibration amplitude represents the temperature factor of the surface, and the pressure strength represents the penetration of the finger into the object. This haptic rendering

concept allows molecular simulations to render touches to molecular surfaces to allow users to feel the dynamics of objects.

- **Performance bottlenecks in most existing VR molecular systems have been identified.** The existing VR molecular system can only simulate the physics of a limited number of objects, albeit only present a rigid object. We identified that the limitation was caused by using only the CPU for the physics engine. Large number of 3D objects will easily overwhelm the CPU physics engine with physics calculations. GPU-based physics engine provides vast number of processing units to compute large number of object physics simultaneously. Therefore, the number of objects that GPU-based physics simulation can handle can be scaled up by adding GPU resources.
- **Determine the most suitable simulation framework for large-scale biomolecules VR simulation.**

  The DirectX 12 particle simulation system uses GPU computing for its physics simulation and has been shown to be able to simulate vast number of particles in VR performance. These particles are the building blocks of molecular 3D objects that can present elastic objects. The object creation method is needed to create objects.

3.11 Significances

The results in this chapter recommend using a simulation framework that utilizes GPU computing as a physics simulation engine. Its ability to simulate large number of objects is an important feature in a high-performance haptic VR system. The DirectX 12 particle simulation system used in this thesis provides a GPU-based physics engine, a hand user interface with haptic feedback, and implemented VR technology. Hence, it is the most suitable simulation framework for large-scale biomolecular VR simulations.

The haptic rendering concept proposed in this thesis provides a way to interpret the tactile sense of touching a molecular object. It can be used in the development of haptic molecular simulations to present a more intuitive interaction with molecular objects using three haptic parameters: vibration frequency, vibration amplitude, and pressure strength.

3.12 Problems and limitations

The development of the natural user interface and the VR simulation engine for the interactive haptic VR simulation platform has encountered some problems which are limitations in the current implementation. These limits are explained as follows:

- **Hardware limitations on haptic implementations.**

  Two pieces of hardware in the haptic implementation whose limitations are considered are the Leap Motion controller and the haptic rendering device. The issues with the Leap Motion controller are related to the accuracy of the tracking. The field of view of the Leap Motion controller is a device-centered inverted pyramid with a typical viewing angle of 140º×120º. If the user's hand is not in this field of view, the controller cannot see the hand and the hand tracking is lost. If that happens, no data will be generated by the controller. Without data received from the controller, the virtual hand will stop moving. Therefore, the user should always place his hands in front of the controller to allow hand interaction.

  Another issue is that the optical sensory devices generally face consistency issues in accuracy regarding viewing angle and distance. Ideally, the object being tracked should be as close as possible and positioned as center as possible of the device's line of sight. The Leap Motion controller is also reported to have varying accuracy according to the location of the hand relative to the controller's sensory space [9]. Therefore, interactions with objects must be managed in the center of the controller's display area for best trace accuracy.

  This issue is related to the haptic rendering device. Compared to the vibrator component for generating tactile feedback vibrations, the servo motor component for pressure feedback has a much slower response. Due to the physical mechanics of the components, the servo motor also produces more noise. Therefore, any hardware that uses a servo motor as an actuator cannot have a high refresh rate. Consequently, pressure feedback on haptic rendering tools needs to be used wisely without having to change the value too often. Creating an efficient haptic algorithm for pressure feedback is one way to get comfortable performance. For example, setting the refreshing pressure parameter to be less frequent to reduce the noise.

- **Limited hand interaction force with 3D objects.**

  In DirectX particle simulation, virtual hands are created from particles. The interaction between hands and objects is determined by the interaction between hand particles and object's particles, which is the collision force that occurs between them. The collision force between particles has limited strength; therefore, the interaction force between hands and objects is also limited. The positive effect is that there is no excessive force from the hand on the object which can turn into an uncontrolled movement because there is practically nothing against the movement of the hand. However, the negative effect is that the ability to move and manipulate objects is also limited. To reduce this deficiency, the collisions

between hand particles and object's particles must be set to be stronger than normal particle collisions but not excessive. Enlarging the hand particles can also help in these interactions.

- **Reduced accuracy for large time step in Euler integration.**
  Euler integration calculates and accumulates all forces every time step. If the time step is large, the calculation may lose some of the details that occur between time steps. Missing details may or may not be important to the simulation results, but it certainly reduces the accuracy of the simulation. As calculations are repeated during the simulation, the impact of the lack of accuracy can accumulate and become significant for some simulations. It is important to have the calculation time step small enough for simulation accuracy. One attempt to obtain a small calculation time step is to apply multiple calculation time steps per graphical time step. That way, the number of compute cycles per second can be in the hundreds even though the graphics frame rate is only 90 Hz.
- **Slow force propagation in particle simulation systems.**
  In a particle simulation, the number of atoms in a molecular object - which means the number of particles in the object - directly affects its mobility. The forces in a particle are propagated to each connected particle in one computing cycle before they are propagated to the next connected particle in the next computing cycle. Since the forces are only propagated over one connection in each computation cycle, more particles result in longer force propagation across the object. Long propagation times prevent objects from smooth subsequent motions. While it reflects the weight of the object, ease of user interaction is also important. The tensegrity representation method that we developed in this study will help the forces to be propagated. This creates a network of spring connections to connect all the object's particles together which will speed up propagation. With this method, the simulated object is easier to move and rotate.

Chapter 4

Tensegrity representation method

4.1 Background

Every 3D object in the particle simulation system is created using particles. To display tangible molecular 3D objects, an object creation method for a particle simulation system needs to be developed. Efficient use of particles to create objects should be one of the factors considered to achieve VR performance.

Biomolecular filament objects were chosen to be presented in this study because most of the active matter objects that generate global dynamic patterns are filament objects such as microtubules. In addition, filament objects are easier to measure their mechanical properties such as elasticity and viscosity. Studies on the measurement of the elasticity of biomolecular filament objects have also been widely carried out; therefore, data on them are easier to obtain for evaluation.

4.1.1 Existing object creation methods

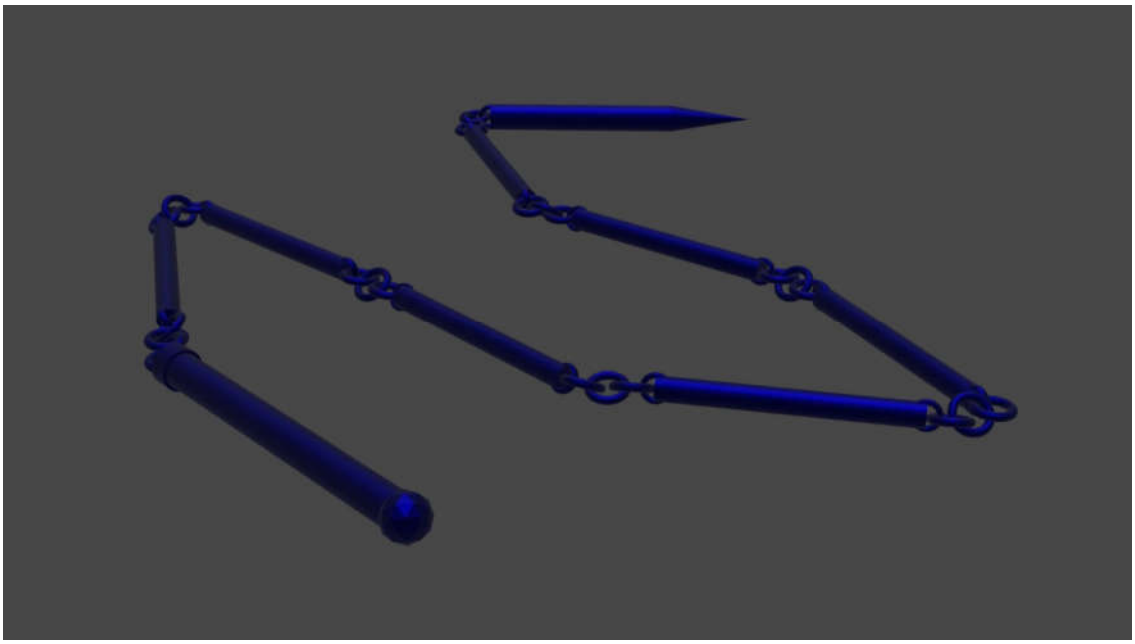
The original particle simulation system in this study uses a spring to connect each particle with the surrounding particles to form a compound object structure [5]. To simplify the discussion, this object creation method will be named as the spring compound method for the remainder of this thesis. This method relies on the structural formation of particles to form the rigidity of objects. As a one-dimensional distance constraint, springs react best to the movement of any connected particle on only one axis. The other movement direction of the connected particle results in a smaller reaction magnitude that varies with the axial angle. Closer to the perpendicular gives the reaction closer to zero.

Since the object rigidity in this method is mainly determined by the structure of the object's particles, it is limited by the shape of the object. Objects without geometric stiffness such as a series of particles is difficult to become rigid with this method (Fig. 15). Increasing the radius of the spring to connect the particles around it helps increase the stiffness, as does increasing the spring constant. However, both approaches have limitations as too much spring forces can cause overshoot which can destroy the object. For long filament objects, this method fails to

propagate any force applied to the entire object's particles, which results in incorrect object bending behavior. Therefore, this object creation method can only present objects with a small degree of rigidity.

Two other object creation methods that we discuss in this chapter are the As-Rigid-As-Possible (ARAP) method with oriented particles [52] and Unified particle physics [53][56]. In fact, these two methods have never been used for molecular simulations. However, since they are both methods for creating objects in particle simulation, we consider them to be potential methods for this purpose.

A



B

Figure 15: Objects without geometric stiffness.
(A) A series of particles that form a compound 3D object is an example of an object without geometric stiffness. (B) A chain weapon as an analogy to the 3D object in Figure A. If the joint between any two connected sticks were particles, and the sticks were springs, then no matter how strong and rigid the sticks were, the object would never be rigid.

The ARAP method can create objects with a wide variety of elasticity. It can even create objects that can withstand extreme deformation and return to their original shape after being

released from stress. However, according to the report, all objects created using this method are very flexible objects. No "almost rigid" object is shown in the results. Furthermore, this method does not use the GPU for physics calculations. As a result, it has a performance bottleneck in collision detection process. Therefore, this method is not scalable and cannot simulate large number of objects in the VR interface.

The Unified particle physics method computes physics using the GPU and has no performance bottlenecks in collision detection. This method can create various types of objects such as gases, liquids, deformable solids, rigid bodies, and cloth. However, this method aims for a sensible visual simulation. The object physics are not designed for accurate simulations. Solid objects created by this method basically are not elastic objects. A deformable solid is a solid that can change shape due to any impact but cannot return to its original form. Elastic objects occur because rigid objects cannot become completely rigid. Object elasticity occurs in these uncontrollable rigid objects. To create real elastic objects, this method must use a spring. Therefore, just like the compound spring method, the formation of particles affects the object rigidity.

Based on the description of the existing methods, there are three issues related to object creation on the interactive haptic VR simulation system platform developed in this study:

- Some methods do not compute physics with the GPU; therefore, it cannot run large-scale simulations.
- The elasticity of objects from the existing methods cannot be widely adjusted.
- Existing methods cannot present rigidity to objects without geometric stiffness.

4.1.2 Needs for a tangible object method

Based on the aforementioned issues, a method for creating tangible objects is needed for the realization of an interactive haptic VR simulation platform. The requirements for the object creation method are as follows:

- **The object creation method must be efficient and parallelable for GPU computation to perform.**

  The performance requirements for an interactive haptic VR simulation platform are to simulate large number of biomolecular 3D objects and fulfil the VR performance requirements of dual graphics rendering in a 90 Hz framerate. To achieve these, the 3D objects in the scene must use the minimum possible number of particles efficiently and the computation load must be properly paralleled for processing on the GPU. To design an

efficient particle object, the focus is on the shape of the object which must match the shape of the biomolecule. Simple objects that ignore particle structure mechanisms usually meet this criterion. For a biomolecular filament object, a series of particles is the most efficient particle used in 3D objects.

- **The object creation method must be able to create objects with widely adjustable elasticities.**

  Natural molecular phenomena involve many different biomolecular objects, each of which may have different mechanical properties. The general property that distinguishes biomolecular objects is their elasticity, especially for biomolecular filament objects. Another factor affecting the visual elasticity of objects is the simulation time step. An object in a smaller time step should move more slowly in the simulation than in a larger time step. The time step affects the calculation of the force magnitude in the simulation, as well as the speed of motion. The object creation method must be able to accommodate the need to create different kinds of object elasticity and visual elasticity of different objects in different time steps.

- **The object creation method must be able to provide rigidity to the object without geometric stiffness.**

  It has been mentioned that efficient particle objects usually ignore the structural formation of the particles. However, the formation of the object's particles greatly affects the stiffness of the object. Since some objects do need to be rigid in simulations, the object creation method must be able to provide rigidity to any object it creates regardless of the structural formation of the object's particles. With an elasticity that does not depend on the particle formation, objects without geometric rigidity can still have rigidity.

4.1.3 Objectives

Derived from the issues and needs of the object creation method already mentioned, the objectives of this chapter are defined as follows:

- Applying the tensegrity structure concept to efficiently create viscoelastic biomolecular 3D objects with particles.
- Adjust the elasticity of the object for a wide range of elasticity values.
- Apply rigidity to objects without geometric stiffness.

In this thesis, we propose a novel unified particle object creation method to fulfill these objectives. We call this method the Tensegrity Representation method because it uses the

principle of tensegrity to form objects and balance the forces that surround them.

4.2 Design considerations

The object creation method on this VR simulation platform must be optimized for performance. Considering that this system will run on a VR performance demand while simulating large number of biomolecular objects, those objects must be a coarse-grained model ensuring parallel computations can be run on the GPU. The efficient use of the number of particles to make biomolecular objects certainly helps its performance. There is no limit to the particle size and no obligation to create a uniform diameter for all particles.

The Tensegrity Representation method utilizes a network of springs to create a geodesic tensegrity object to hold the object's particles in the shape. It is common to use springs to create flexible 3D objects, but also known to be very challenging. Incorrect setting of the spring constant can cause a force imbalance that results in erratic bounce motion of the object and the explosion of objects. Longer or larger objects require more particles and springs which increase the risk of overshoot. Therefore, maintaining the stability of objects is an important consideration in design.

The biomolecular 3D objects in the simulation are expected to have mechanical behavior that makes sense according to the results measured in the laboratory experiment, although they may not match them perfectly. Several measurement criteria have been prepared to evaluate the object, such as bending shape, viscoelastic behavior, and adjustment for flexural rigidity.

4.3 Particle simulation system for Tensegrity Representation

This study presents a VR particle simulation system derived from previous work on the GPGPU accelerated microtubule gliding assay simulation system [12][13][14] and a haptic rendering system for VR molecular modeling [10][11]. The movement of tens of thousands of particles is simulated in this VR system to reproduce the targeted biomolecular phenomena. Each force is calculated to be the particle's additional velocity or acceleration at each time step. Euler integration is used to integrate forces to move particles (Fig. 14). The drag (damper) value is applied to all particles in the system to reduce all particle velocities per computation cycle. The velocity of all particles is reduced by this multiplier value for each computing cycle creating the impression of a viscous solution surrounding the particles. The particle's velocity and new location is described as:

$$Velo_n^i = Velo_n^{i-1'} \times \left(1 - k_{drag}\right), \quad (4)$$

$$Loc_n^{i+1} = Loc_n^i + Velo_n^i, \quad (5)$$

where $Velo_n^i$ is the n^{th} particle velocity in the current compute cycle (i), $Velo_n^{i-1'}$ is the n^{th} particle velocity from the previous compute cycle (i-1) with additional forces, k_{drag} is the drag value, Loc_n^i is the n^{th} particle 3D location (position) in i compute cycle, and Loc_n^{i+1} is the n^{th} particle 3D location (position) in the next compute cycle (i+1).

Collisions between particles are detected and determined by the penetration of two particles. It is calculated by the distance between the centers of the two particles and the sum of the radii of the two particles. The repulsive forces that push both directions are generated when penetration occurs. A deeper penetration causes a greater repulsion. The different diameters of the two different impact particles do not pose a problem and can be applied well. However, with the current default, each particle is set to have the same mass so that the resulting repulsive forces for the two colliding particles are set the same. The velocity of the bouncing motion for each particle is then determined by the initial velocities and the repulsive force.

The collision force is calculated individually per particle for every computing cycle. This allows computation tasks from collision processes to be processed in parallel using multiprocessing resources such as multicore CPU and multiple GPUs. Originally, the interactions between particles were calculated using the Lennard-Jones potential formula, which calculates repulsive and attraction forces. However, for the current purposes, only the repulsive force is required as the collision force in the simulation. Therefore, the attractive force in the formula is negated in this simulation as shown in Figure 16.

The springs are implemented by providing each particle with three lists of data: a list of connected particle IDs, a list of the default distances of the connections, and a list of each spring constant. Which particle to connect to for each particle is determined by the object creation method. The distance between a pair of connected particles can change due to the particles' motion, creating a difference with the default distance. Multiplied by the spring constant, it becomes the Hooke's Law equation used to calculate the spring force. Just like collision calculations, this spring calculation can be calculated simultaneously in parallel by using multiple GPUs or multicore CPU.

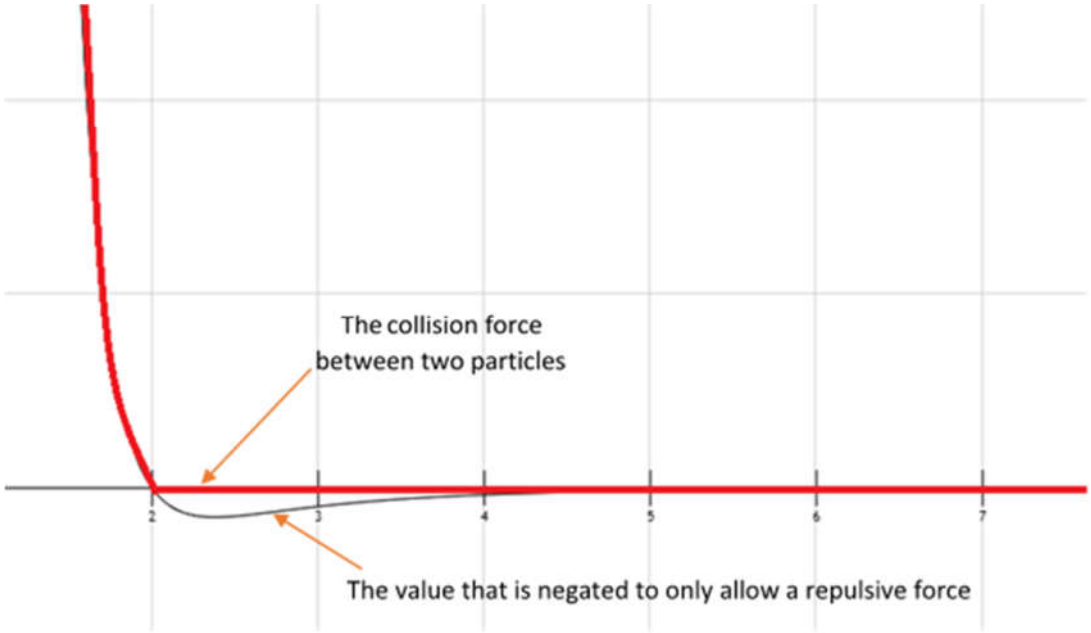


Figure 16: Illustration of collision calculation formula.
Derived from the Lennard-Jones potential formula, the attractive forces are removed from the result to allow only repulsive forces.

In the tensegrity representation method, the object's particles are formed by a network of springs and anchors. An anchor is an abstract 3D object that is almost identical to a particle. An anchor has location coordinates and can be attached by springs to other anchors or particles. As an abstract 3D object, it is invisible and does not collide with particles or other anchors. Anchors are placed around the object's particles and binds them with springs to maintain the object's shape. There are data lists stored for anchor-to-particle connections and anchor-to-anchor connections; therefore, the calculations can be carried out in parallel. By ensuring this parallelism, all computations regarding anchors can be processed simultaneously using multiple GPUs or multicore CPU.

The forces acting on each particle are the collision force ($F_{collision}$), the spring force from the connected particles ($F_{particleToParticle}$), and the spring force from the connected anchor ($F_{anchorToParticle}$). The overall force at each particle ($F_{particle}$) is determined by the sum of all the forces acting on it as:

$$F_{particle_n} = \sum_{i=1}^{a} F_{collision_i} + \sum_{j=1}^{b} F_{particleToParticle_j} + \sum_{k=1}^{c} F_{anchorToParticle_k} \tag{7}$$

The forces acting on each anchor are also defined almost the same as on each particle. However, there is no collision force against the anchor because it does not collide with any object. The overall force at each anchor is defined as:

$$F_{anchor_n} = \sum_{i=1}^{f} F_{particleToAnchor_i} + \sum_{j=1}^{g} F_{anchorToAnchor_j} \tag{8}$$

where the spring forces from the connected particles is $F_{particleToAnchor}$ and the spring forces from the connected anchors is $F_{anchorToAnchor}$.

The overall spring force acting on each particle and anchor is greatly influenced by the placement of the anchor and spring, which depends on the object creation method. The Tensegrity Representation method carefully establishes the anchors and springs of the object it creates to achieve a force balance while providing the elasticity of the object.

4.4 Tensegrity representation object model

Microtubule objects in the microtubule gliding assay simulation [12][13][14] are presented as a series of particles connected by springs. Each particle represents a segment in the microtubule body with a length and diameter of 24 nm, which corresponds to the microtubule diameter. Other rod-like macromolecular objects such as actin filaments, DNA strands, collagen can also be represented in the same way, only with a different size scale. It is difficult to give rigidity to objects of this kind using only the springs between the particles due to the absence of the geometric stiffness. The Tensegrity representation method creates objects by placing anchors and springs around the object's particles to form a tensegrity structure and support its conformation (Fig. 17). The anchor is analogous to the surrounding solution that interacts with the object through the attached springs.

The Tensegrity representation object is classified as a geodesic tensegrity structure because the strength of the structure is formed by a network of springs that can produce both tension and compression. The spring as a structural element to connect two particles or anchors serves as a distance constraint which produces a pulling force when the distance is farther and produces a pushing force when it gets closer. This structure can withstand any external pressure that deforms its shape and returns to its original shape after the pressure is removed. This ability makes the object elastic while giving it a rigid strength.

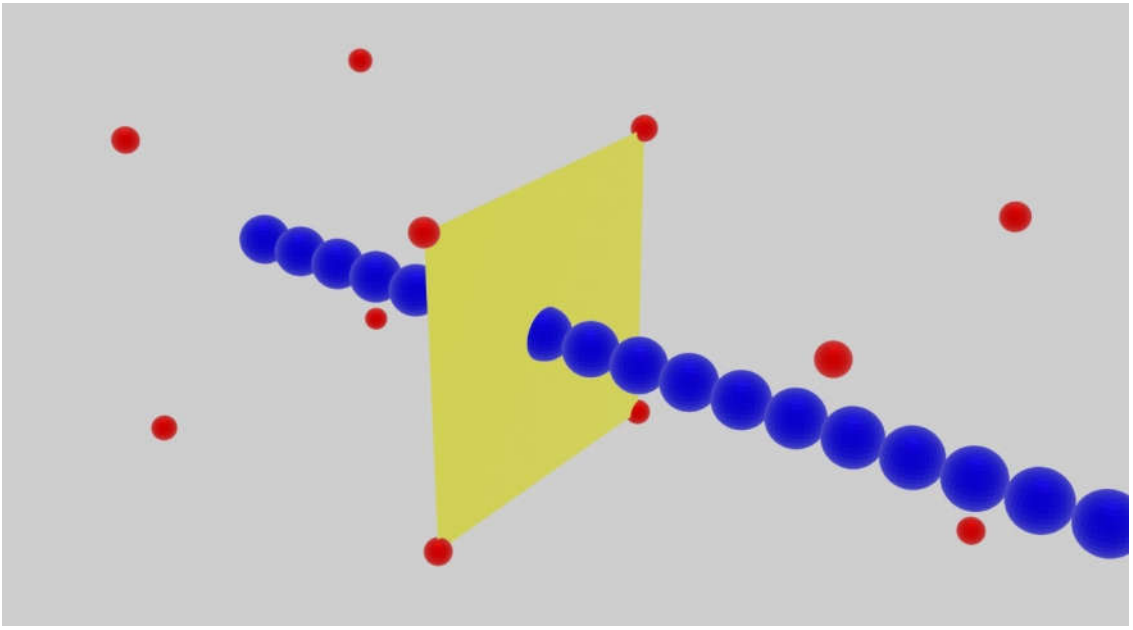


Figure 17: The formation of anchors and particles in the Tensegrity representation object. Blue spheres represent the object's particles, red spheres represent the anchors, and a yellow square represents a rectangular formation of anchors perpendicular to the object's axis. Red spheres and yellow square are abstract objects that are not visible.

The Tensegrity representation method arranges sets of four anchors along the length of the object from one end to the other. Each of the four anchors is placed in an imaginary rectangular plane perpendicular to the axis of the filament object, forming a square formation. Each of these sets of anchors are connected to each other by strings to form a tensegrity structure and maintain its shape. The object particle that lies between two sets of four anchors is called a section. They are held by two sets of four anchors which flank them. Illustration of spring connections in this arrangement is depicted in Figure 18.

The basic calculation of the spring force is Hooke's Law. It defines as a spring constant times the changing distance between two connected or anchored particles (Δd). The changing distance Δd is defined as the difference between the current distance d and the initial distance d_{init}. The equations of spring force in the Tensegrity representation method are defined as:

$$F_{particleToParticle} = k_{pp} \times \Delta d \,, \tag{9}$$

$$= k_{pp} \times (d - d_{init}) \,, \tag{10}$$

$$F_{anchorToAnchor} = k_{aa} \times \Delta d \,, \tag{11}$$

$$= k_{aa} \times (d - d_{init}) \,, \tag{12}$$

$$F_{anchorToParticle} = k_{ap} \times \Delta d \,, \tag{13}$$

$$= k_{ap} \times (d - d_{init}) \,, \tag{14}$$

$$F_{particleToAnchor} = k_{pa} \times \Delta d \,, \tag{15}$$

$$= k_{pa} \times (d - d_{init}) \,, \tag{16}$$

$$k_{pa} = \frac{1}{numberOfParticles} \,. \tag{17}$$

In Eq. 9 and 10, the spring force of two connected particles $F_{particleToParticle}$ has the same spring constant k_{pp} for both particles. The spring force of two connected anchors $F_{anchorToAnchor}$ in Eq. 11 and 12 has also the same spring constant k_{aa} for both anchors. However, the spring force between a particle and an anchor is different. The spring force on the particle $F_{anchorToParticle}$ in Eq. 13 and 14 uses the spring constant value k_{ap} set by the object parameter, whereas the spring force at the anchor $F_{particleToAnchor}$ in the Eq. 15 and 16 has a spring constant value k_{pa} as one per number of particles connected to it (*numberOfParticles*) in Eq. 17. This approach is used because the anchor may connect to large number of particles when the object created is large. The risk of overshoot that can cause the object to explode is high as the anchor might receive too much spring force. The excessive spring force can be avoided by setting the spring constant with the number of connected particles as a divider. It seems that the different spring constants used between the pair of spring forces of the anchor and the particle creates uneven forces. However, the two spring forces will balance each other by the Euler integration in the next time steps. Therefore, the stability of the object is maintained.

For very rigid biomolecules such as microtubules, presenting this level of rigidity is challenging. Setting the spring constant too high is prone to object overshoot. To avoid excessive forces, the Tensegrity representation method implements bridge connections between anchor sets. This helps to make the object more rigid while at the same time speeding up the propagation of any force by connecting each anchor set to the other. Bridge connection 0 setting does not establish any bridge connection between anchor sets. The bridge connection 1 setting adds additional connections for each anchor with adjacent anchors in a row. The bridge connection 2 setting creates the connection between each anchor and the next seconds anchor in a row, and so on for the subsequent bridge connection settings (Fig. 18).

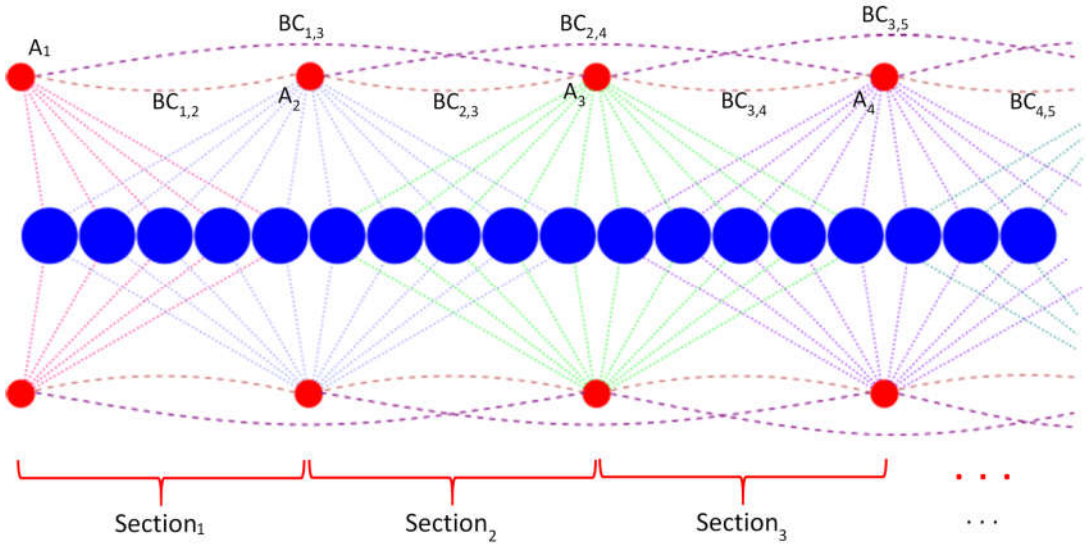


Figure 18: The Tensegrity representation object connection scheme illustrated in 2D. Each anchor set connects to all particles in two adjacent sections. In bridge connection 0 setting, no connection between anchor sets is provided. In bridge connection 1 setting, anchor A1 is connected to anchor A2 (BC1,2), anchor A2 is connected to anchor A3 (BC2,3), anchor A3 is connected to anchor A4 (BC3,4), and so forth. In bridge connection 2 setting, anchor A1 is connected to anchor A3 (BC1,3), anchor A2 is connected to anchor A4 (BC2,4), anchor A3 is connected to anchorA5 (BC3,5), and so forth.

The arrangement in the Tensegrity representation method raises seven object parameters that can be adjusted to produce object elasticity. These parameters are:

- Particle-to-particle constant.
- Anchor-to-anchor constant.
- Anchor-to-particle constant.
- Anchor distance, which is the distance between each anchor to the center of the anchor set.
- Number of sections in the object.
- Section length, which is the number of particles per section.
- Bridge connection.

Only the anchor distance and section length parameters can have a unit of measure (meters). How long these two parameters are in meters depends on the actual length of the particle diameter in meters.

4.5 Evaluations

The Tensegrity representation object was evaluated to fulfill the objectives of creating

tangible objects. Objects were evaluated for elasticity, bending shape, and viscoelastic behavior. As is usually done in experiments, elasticity is measured by flexural rigidity. The methods for evaluating the Tensegrity representation object are described in the following sections.

4.5.1 Static flexural rigidity measurements

Flexural rigidity is defined as the resistance of an object when it is bent by external forces. The symbol EI to denote comes from E for the Young's modulus and I for the second moment of area. The significant difference in the flexural rigidity values observed in several works occurred because each method used involved different observational features. Methods that use dynamic features such as hydrodynamics involve more uncontrolled forces, so they are not easily measured. According to Kikumoto et al. [19], more precise and consistent results can be obtained with more static and more direct methods.

This study directly measures the flexural rigidity of the Tensegrity representation object using classical mechanical analysis. The molecular filament object in this measurement was assumed to be isotropic, as previous work also assumes. The method of measurement was done by bending the object and analyzing the results. An object with length L is bent by keeping one end in fixed position and orientation, then moving the other end perpendicularly upward with a certain force F_{up}. Fixing the ends of the object is done by fixing the position of the four anchors at these ends. The force F_{up} is applied by moving the end particle with mass m gradually upward with acceleration a as:

$$F_{up} = ma \, . \tag{18}$$

The acceleration a is defined as:

$$a = \frac{V}{t} = \frac{r}{t^2} \, , \tag{19}$$

where V is the velocity added to move particle, t is the time for one computing cycle, and r is the moving range of the particle in each computing cycle.

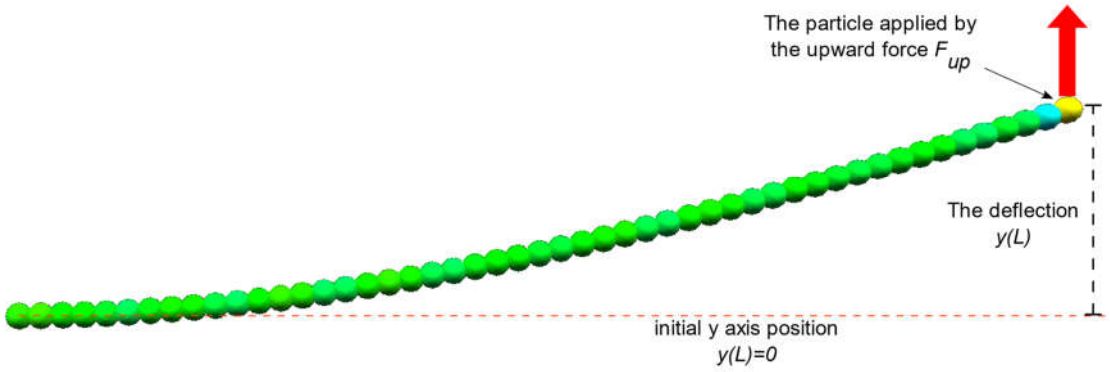


Figure 19: Static flexural rigidity measurement for the Tensegrity representation object. The object bends to a stop slowly in a stable state after no further increase in deflection. The deflection of the particles under steady state conditions was analyzed. The maximum deflection of the moving end $y(L)$ determines the flexural rigidity.

Figure 19 shows the bending of the object. It bends slowly until the force-balance is reached and the bending stops in a stable state. The endpoint of the object in the actual measurement simulation never completely stops moving because there is a small fluctuation in the maximum deflection. The condition is assumed to be in a steady state when the maximum deflection of the endpoint of object does not increase in about 100-300 computation cycles. Therefore, the maximum deflection $y(L)$ is determined by the mean value of these first 100 - 300 computation cycles (Fig. 20). The value of flexural rigidity is obtained using classical mechanics bending of rods equation as:

$$\mathrm{y}(L) = \frac{F_{up}L^3}{3EI}, \tag{20}$$

$$= \frac{mrL^3}{3t^2EI}, \tag{21}$$

$$EI = \frac{mrL^3}{3t^2y(L)}. \tag{22}$$

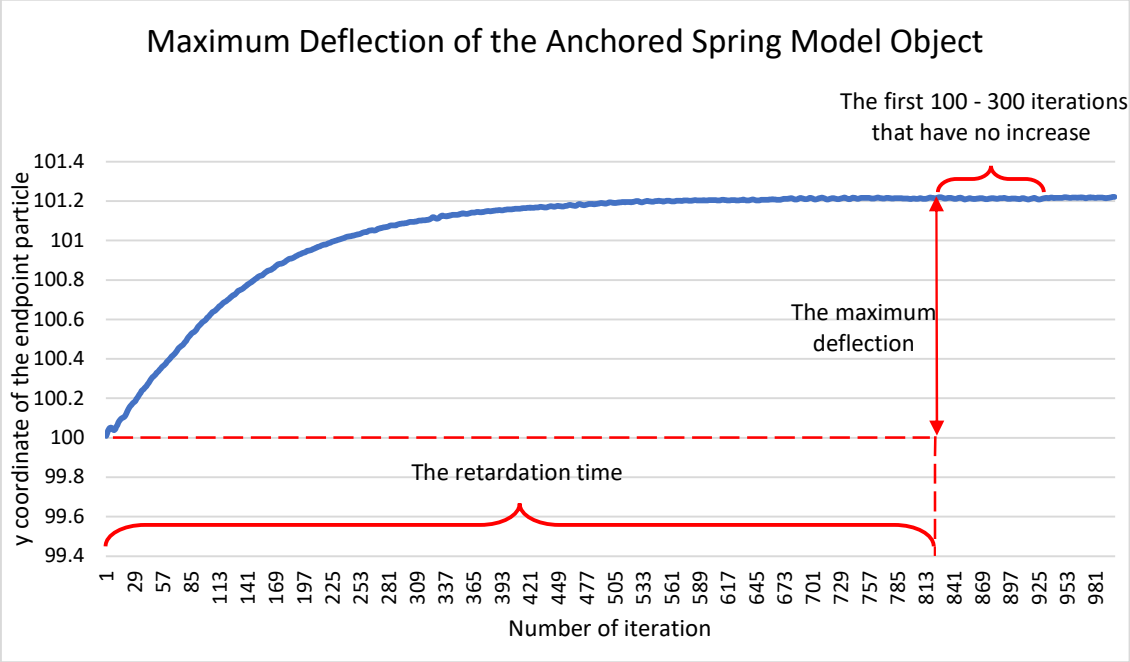


Figure 20: The maximum deflection of the Tensegrity representation object.
It is measured by the mean value of the endpoint particle deflection in the first 100 - 300 computation cycles that do not increase.

In Eq. 22, all variables except the flexural rigidity *EI* and the maximum deflection $y(L)$ are bound to the type of molecular object represented. The mass of a particle m is determined by the molecular object it represents. The moving range of the particle r in the object's end and the length of the object L depends on the real value of the particle's diameter it represents. Time t is derived from the simulation time scale t_{sim} divided by the number of computing cycles per second *cps*:

$$t = \frac{t_{sim}}{\text{cps}}\,. \tag{23}$$

To make the flexural rigidity EI apply generally to any molecular object, Eq. 22 can be rearranged to separate object-specific variables and simulation variables as:

$$EI = \frac{n_{particles}\frac{r_{sim}}{d_{sim}}\left((n_{section}\times section_{length})-1\right)^3}{3y(L)} \times m_{particle} \times {d_{particle}}^3 \times \frac{cps^2}{{t_{sim}}^2}\,, \tag{24}$$

where $n_{particles}$ is the number of moving particles in the object's endpoint, r_{sim} is the moving

range of the moving particles in unit length of simulation, d_{sim} is the diameter of a particle in unit length of simulation, $n_{section}$ is the object's parameter that determine the number of section in the object, $section_{length}$ is the object's parameter that determine the number of particles per object's section, $m_{particle}$ is the mass represented by a particle in kilogram, and $d_{particle}$ is the diameter represented by a particle in meter. The result values from the simulation are the simulated flexural rigidity EI_{sim}:

$$EI_{sim} = \frac{n_{particles}\frac{r_{sim}}{d_{sim}}\left((n_{section}\times section_{Length})-1\right)^3}{3y(L)}. \quad (25)$$

By adding Eq.25 into Eq. 24, EI becomes:

$$EI = EI_{sim} \times m_{particle} \times {d_{particle}}^3 \times \frac{cps^2}{{t_{sim}}^2}. \quad (26)$$

By placing EI_{sim} from Eq. 20 to the left and other variables to the right, the relationship between EI_{sim} and EI is shown by:

$$EI_{sim} = \frac{EI}{m_{particle}\times {d_{particle}}^3 \times \frac{cps^2}{{t_{sim}}^2}}. \quad (27)$$

4.5.2 Bending evaluation

The bending shape of the Tensegrity representation object is analyzed by comparing the actual deflection of each object's particle with the calculation of the classical mechanical bending equation. This evaluation is used to ensure that the Tensegrity representation object bends properly according to the physical bending law. The deflection of each particle along the object varies depending on the distance x from the fixed endpoint of the object. According to classical bending mechanics, the expected deflection along the object is obtained as:

$$\mathrm{y}(x) = \frac{F_{up}L^3}{6EI}\left[-\left(\frac{x}{L}\right)^3 + 3\left(\frac{x}{L}\right)^2\right]. \quad (28)$$

The bending shape of the Tensegrity representation object is evaluated by comparing the

y-axis position of each object's particle with the expected deflection according to Eq. 28. A result which is close to the expected deflection means that the bending of the object follows the classical law of bending.

4.5.3 Viscoelastic behavior

Viscoelastic behavior is investigated by observing the time it takes for the object to reach a steady state in a bending process called retardation time. In this measurement, the retardation time can be determined from the number of computing cycles required for the object's endpoint deflection to stop or not to increase significantly (Fig. 20).

The viscoelastic behavior of the object is expected to be the same as the Kelvin (Voigt) rheological model with increasing length of the object resulting in an increase of the number of viscous dashpots in the model. This means that a longer object has a longer time to reach a steady state. The movement in the retardation and relaxation processes is also expected to follow the creep-recovery response of the Kelvin (Voigt) rheological model.

4.6 Results

The molecular object represented in this measurement is microtubule. One object's particle represents a segment in the microtubule that contains three cycles of tubulin dimers. This is equal to the microtubule diameter of 24 nm, which is represented as the object's particle diameter. In this measurement simulation, the object's particle diameter is one unit of simulation length. Therefore, 24 nm is multiplied to get the real-world value of the anchor distance and the section length parameters. The mass of a particle is $6.4761023597\times10^{-21}$ kg which comes from 78 monomers in one microtubule segment represented by one particle. The computing cycle is assumed to be the same as the graphics framerate, which is 90 Hz for VR performance. The simulation time scale is 1: 4×10^{-6}, which means one second in simulation is equal to 4 microseconds in the real-world time. Therefore, one computing cycle time represents 44.44 nanoseconds. The object's length is determined by multiplying the number of sections, the section length, and the particle diameter, then subtracting one particle diameter.

4.6.1 Bending evaluation results

The results of the bending evaluation show the conformity of the bending shape of the Tensegrity representation object with the classical bending equation (Fig. 21). The deflection of each particle in the measurement simulation is almost the same as the deflection expected

from the calculation using Eq. 28. These results prove that the Tensegrity representation object can be bent properly according to the laws of mechanics. This result is possible because the anchors manage to propagate the bending force from one end of the object to all the particles of the object.

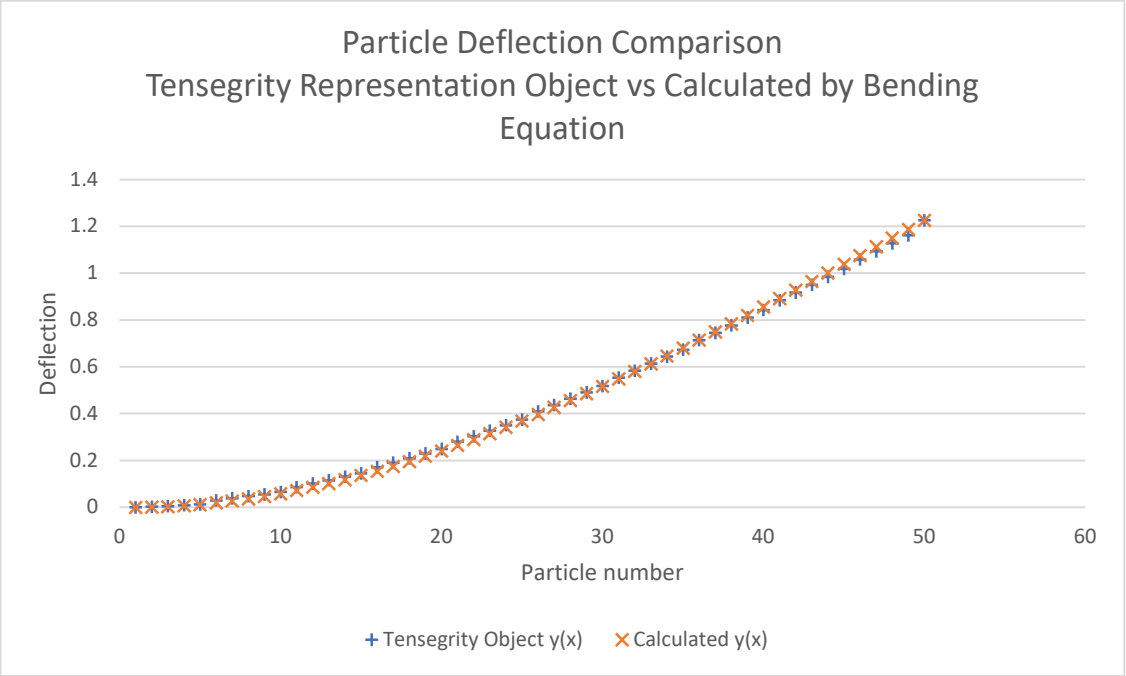


Figure 21: Bending evaluation of the Tensegrity representation object compared to classical bending equation.
The deflection of each object's particle is compared with the expected deflection calculated by mechanical equation. The results show a high conformity of the two, proving that the Tensegrity representation object can bend properly according to the laws of mechanics.

4.6.2 Viscoelasticity observation results

The Kelvin (Voigt) rheological model as a reference for the viscoelastic behavior of biomolecular objects shows an increase in viscosity when the object is longer which causes the object's retardation time to be longer. Two object parameters define the length of the object: the number of sections parameter and the section length parameter. Higher number of sections means more groups of object's particles. Higher section length means more particles for each object's section. The two values are linear with the length of the object, and it is also linear with the retardation time as shown in Table 8 and Table 9.

Table 8: Retardation time of the Tensegrity representation object with a varying number of sections.

Number of sections	Microtubule length (nm)	Retardation time	
		Number of computing cycles	Time (μs)
5	576	153	6.80
6	696	254	11.29
7	816	603	26.80
8	936	989	43.96
9	1056	1712	76.09
10	1176	2581	114.71

Table 8 and Table 9 show the results which indicate that the increasing length of the Tensegrity representation object does makes the retardation time longer, which means that the viscosity increases. This behavior is in line with the Kelvin (Voigt) rheological model found in the experimental results.

Table 9: Retardation time of the Tensegrity representation object with a varying section length.

Section length	Microtubule length (nm)	Retardation time	
		Number of computing cycles	Time (μs)
2	456	363	16.13
4	936	1303	57.91
6	1416	3136	139.38
8	1896	5217	231.87
10	2376	7446	330.93

The object was further tested by analyzing the movement of the object's moving end in the retardation and relaxation process to compare it with the creep-recovery response of the Kelvin (Voigt) rheological model [65] (Fig. 22). The maximum deflection of the object is shown in Figure 23 to illustrate the movement of the object's moving end in the retardation and relaxation

processes. These two figures show the similarity of the object's movement in the retardation and relaxation processes and the creep-recovery response of the Kelvin (Voigt) rheological model. This shows the validity of the viscoelastic behavior of the Tensegrity representation object.

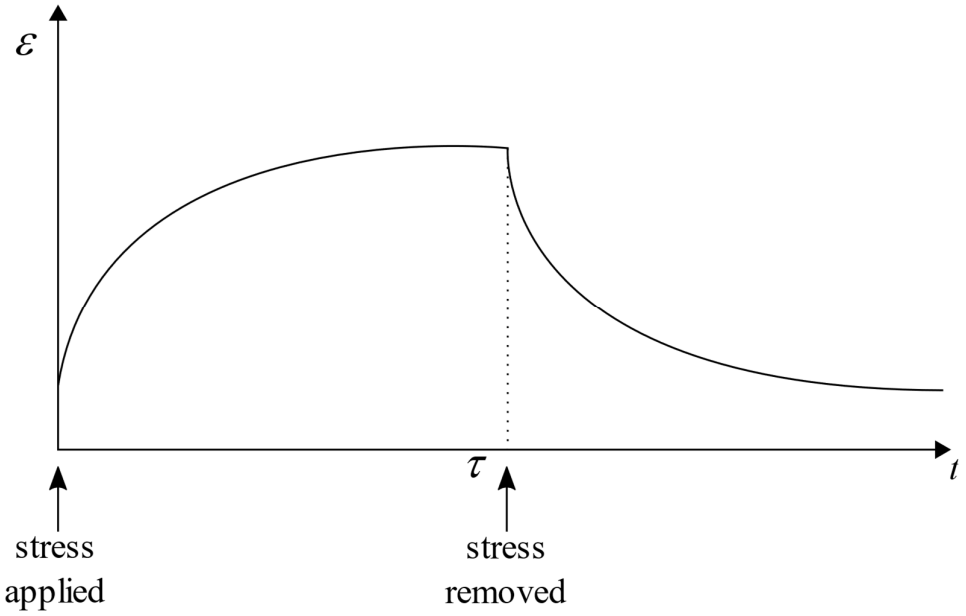


Figure 22: Creep-recovery response of the Kelvin (Voigt) rheological model as a reference in evaluating the Tensegrity representation object.

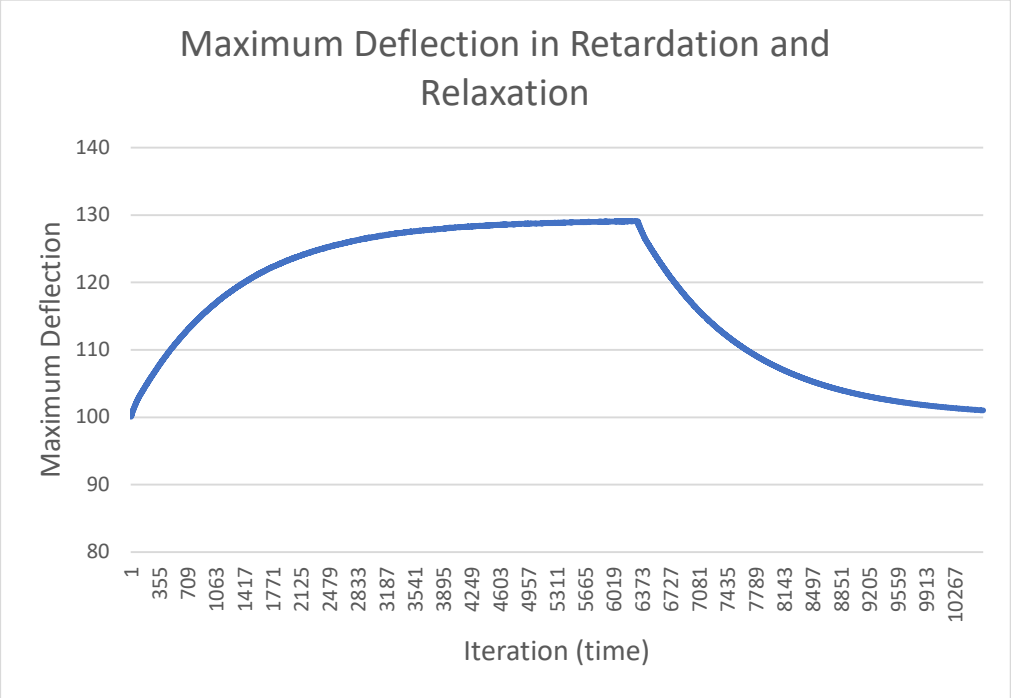


Figure 23: The maximum deflection of the Tensegrity representation object. The endpoint of the object deflects after the bending force is applied, then slowly stops at a certain value (maximum deflection). When the force is removed, the object's endpoint moves to restore its shape and slowly stops to a zero-deflection value.

4.6.3 Flexural rigidity measurement results

The object's flexural rigidity is measured by running a measurement simulation to find the maximum deflection as described in the previous section. Object parameter settings determine the flexural rigidity of the object. Finding the formula for these parameters to obtain the desired flexural rigidity is a challenge. The number of possible combination values derived from the seven parameters is unlimited because of the continuous values of most parameters. The correlation between each parameter with the flexural rigidity of the Tensegrity representation object can be obtained by measuring it with various values of each individual parameter and analyzing the results.

A set of default object's parameters was first defined to find the correlation between the parameters and the flexural rigidity. Flexural rigidity was measured several times by changing the parameter value of each parameter within a certain range of values. For each parameter, a trendline was obtained in correlation with the flexural rigidity. To find a correlation with other parameters, the values of other parameters are changed one by one while repeating the steps

for finding the trendline. Several curves are obtained as a result with each representing a baseline and a correlation with other parameters. The default values, variation values, and modified values used in this study was shown in Table 10.

Table 10: Parameters and values for correlation analysis with flexural rigidity. Default values are the values used in each parameter analysis as the baseline values. The variation values are the values that are changed for each parameter to obtain a correlation curve. The modified values are the values that will be changed in each parameter to get the correlation curve of the other parameters. To get the real unit value, the anchor distance parameter and section length are multiplied by 24 nm.

Parameter	Default value	Variation value	Modified value
Particle-to-particle constant	0.1	0.05, 0.01, 0.15, 0.2, 0.25	-
Anchor-to-anchor constant	0.2	0.05, 0.01, 0.15, 0.2, 0.25	0.15
Anchor-to-particle constant	0.1	0.05, 0.01, 0.15, 0.2, 0.25	0.15
Anchor distance	8	5, 10, 15, 20, 25	15
Number of sections	10	5, 6, 7, 8, 9, 10	7
Section length	5	2, 4, 6, 8, 10	2
Bridge connection	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	-

Figure 24 shows the results as correlation curves for each object's parameter. There are several parameters that show a linear correlation with flexural rigidity. There are also two parameters with power trendline correlations and one with random results. These results are then analyzed to determine the fitting function for regression. The final product of this analysis is the parameter fitting function to obtain the desired flexural rigidity.

The particle-to-particle constant parameter showed no pattern in the results. measurements to get the trendline were carried out three times and the results showed random flexural rigidity values with only small differences in magnitude as shown in Figure 24A. Therefore, it was concluded that this parameter does not have a significant contribution to the value of the object's flexural rigidity and there was no need for further analysis by looking for correlations with other parameters. This result was not surprising because the object being measured was a series of particles which is an object without geometric stiffness. Following these results, the particle-to-particle constant parameter was excluded from the correlation analysis with the

flexural rigidity values.

The anchor-to-anchor constant parameter, the anchor-to-particle constant parameter, and the number of sections parameter show a positive linear correlation with the flexural rigidity (Fig. 24B, 24C, and 24D). The linear correlation curve is described as:

$$\mathrm{y} = m(x + a) + b\ , \tag{29}$$

where y is the flexural rigidity, m is the slope of the curve, x is the object's parameter, a is a constant, and b is also a constant. The m slope of the trendline curve changes in a positive direction with changes in the other object's parameters. Thus, it can be concluded that these parameters are multiplied by other variables in the parameter fitting function.

The anchor distance parameter and the section length parameter are shown in Fig. 24E and 24F with a power trendline in correlation with the flexural rigidity. The trendline curves are defined as:

$$\mathrm{y} = cx^{b}\ . \tag{30}$$

Changing the values of other object's parameters does not change the power constant b; on the contrary, it changes the scaling factor c in a positive direction. This indicates that the relationship between these power trendline parameters and other parameters in the parameter fitting function is also a multiplication.

Bridge connection parameter has the most unique trendline curve of all (Fig. 24G). Initially, it shows a significant increase, but suddenly decrease at a certain value. The presence of bridge connection springs significantly increases the object rigidity. An object with bridge connection 1 setting has nearly twice the flexural rigidity compared to that with bridge connection 0 setting as the bridge connection springs connect each anchor with adjacent anchors. Increasing the bridge connection setting value strengthen the tensegrity structure of the object. The illustration of the bridge connection is depicted in Figure 18. In bridge connection 1 setting, most of the anchors are connected by bridge connection springs to two anchors each — one on the left and one on the right. For example: anchor A_2 is connected to A_1 (with bridge connection $BC_{1,2}$) and A_2 ($BC_{2,3}$); anchor A_3 is connected to A_2 ($BC_{2,3}$) and A_4 ($BC_{3,4}$); and so on. However, there are two anchors at each endpoint of the object have only one the adjacent anchor to connect to. Anchor A_1 is connected to anchor A_2 ($BC_{1,2}$), but there is no anchor A_0 to connect to. At the

other end, anchor A_n is only connected to anchor A_{n-1} ($BC_{n-1,n}$) because there is no anchor A_{n+1}. Therefore, the number of bridge connection springs is n-1. In bridge connection 2 setting, each bridge connection spring skips one anchor in between. Therefore, four anchors — two anchors at each end — have one side of the neighboring anchor to connect to. Anchor A_1 is only connected to anchor A_3 ($BC_{1,3}$), anchor A_2 is only connected to anchor A_4 ($BC_{2,4}$), anchor A_{n-1} is only connected to anchor A_{n-3}, and anchor A_n is only connected to anchor A_{n-2}. Therefore, the number of bridge connection springs is n-2. The reduction continues for the next bridge connection value setting. Increasing the bridge connection value setting decreases the number of bridge connection springs in the object. This creates a kind of trade-off between a stronger structural spring formation and a smaller number of springs. Therefore, at a certain value, the flexural rigidity begins to decrease.

The results showed that the increase in the value of the flexural rigidity stopped when the value of the bridge connection was about one-third of the object's number of sections. Therefore, the bridge connection setting value used in this study is less than one-third of the number of sections value. The analysis of the adjusted bridge connection parameter is shown in Figure 24H when the curves are cut before reaching one-third of each number of sections value. This curve shows the correlation of the bridge connection parameter with the flexural rigidity which is seen as linearly positive (Equation 29). This parameter also indicates that it should be multiplied by other variables in the parameter fitting function for flexural rigidity.

The results curves of this parameter correlation analysis are important to determine the fitting function for regression analysis. Using the Generalized Reduced Gradient (GRG) Nonlinear solving method, the parameter fitting function for finding the flexural rigidity of the Tensegrity representation object is defined as:

$$EI_{sim} = 0.29.(x_1 + 0.04).x_2.{x_3}^{1.7}.(x_4 + 3.27).{x_5}^{2.2}.(x_6 + 0.97)\ . \qquad (31)$$

where x_1 represents the anchor-to-anchor constant parameter, x_2 represents the anchor-to-particle constant parameter, x_3 represents the anchor distance parameter, x_4 represents the number of sections parameter, x_5 represents the section length parameter, and x_6 represents the bridge connection parameter.

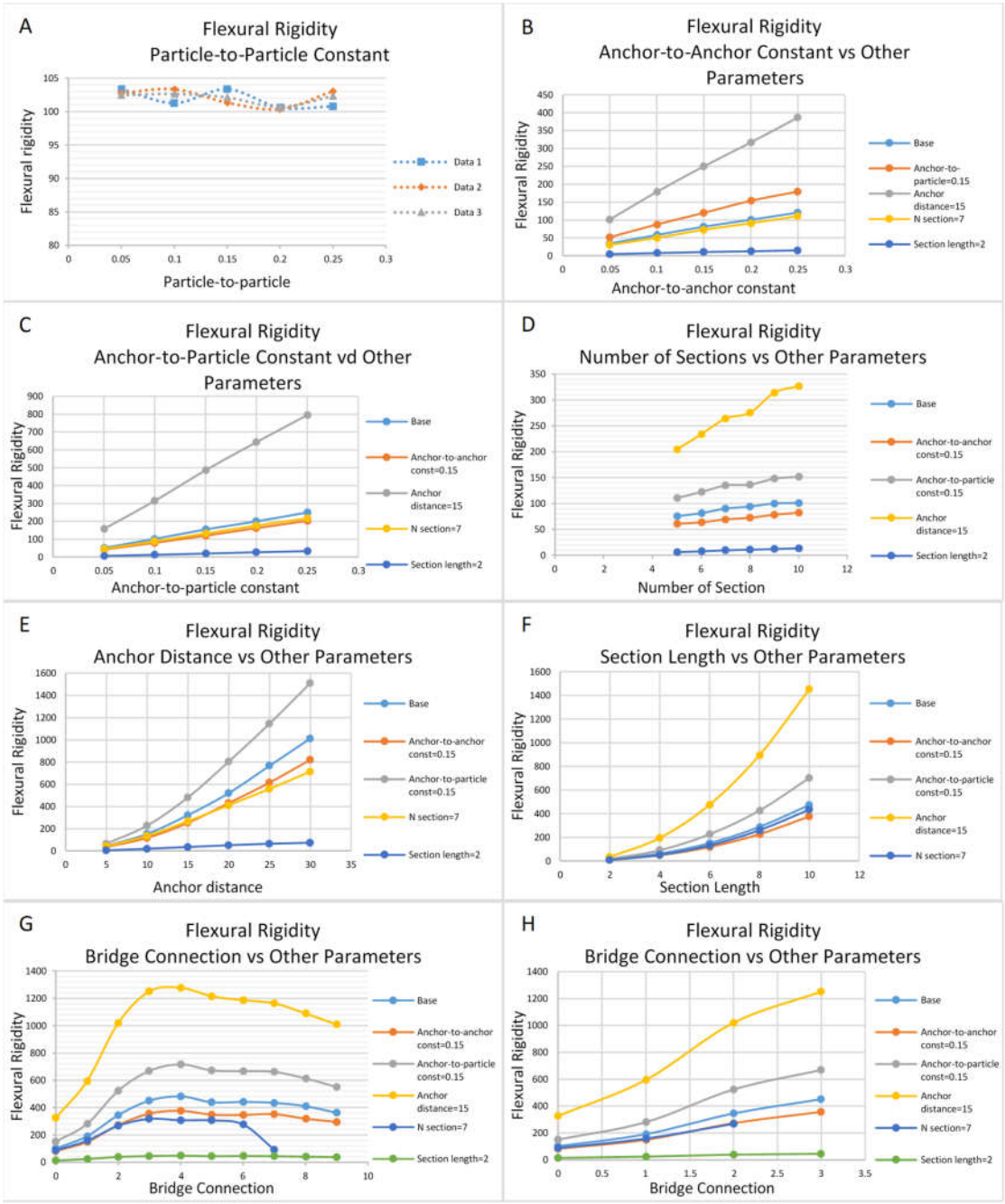


Figure 24: Correlation between object's parameters with the flexural rigidity in the Tensegrity representation.
(A) Particle-to-particle constant parameter. (B) Anchor-to-anchor constant parameter. (C) Anchor-to-particle constant parameter. (D) Number of sections parameter. (E) Anchor distance parameter. (F) Section length parameter. (G) Bridge connection parameter. (H) Adjusted bridge connection parameter.

The parameter fitting function provides a calculation equation for obtaining the object's flexural rigidity. The accuracy of this function is not perfect, but it is close enough to the

targeted flexural rigidity. The actual value of the flexural rigidity should be measured as in the previous section. For a more accurate setting of flexural rigidity, this function is very useful as an initial parameter setting to get closer to the targeted value. Then the parameters can be readjusted to match the desired flexural rigidity value. The parameter that is easiest to adjust is one that has a linear correlation and a continuous value, namely the anchor-to-anchor constant parameter and the anchor-to-particle constant parameter.

The flexural rigidity EI_{sim} produced in the simulated world is intended for general use for any molecular object. To convert EI_{sim} to real-world EI value, the values of the object-specific variables must be defined. Eq. 24 converts the EI_{sim} with additional variables $m_{particles}$ (particle mass), $d_{particles}$ (particle diameter), cps (number of computation cycles per second), and t_{sim} (simulation time scale). For microtubule objects, the $m_{particle}$ is $6.4761023597 \times 10^{-21}$ kg, the $d_{particle}$ is 24 nm, cps is 90, and the t_{sim} is 44.44 nanoseconds.

The parameter fitting function was tested by presenting two microtubule flexural rigidity target values: 3.7×10^{-24} Nm^2 of pure microtubules (GDP tubulin) and 16×10^{-24} Nm^2 of microtubules with MAP. Both values were obtained from the results of experiments using optical tweezers with the RELAX method [45]. The two values were chosen because they came from the same research with the same method but with a significant difference in magnitude. These flexural rigidity values (EI) were converted into the flexural rigidity values of the EI_{sim} simulation using Eq. 27. to 81637.42 for the value of pure microtubules and 353026.7 for the value of microtubules with MAP. Then the parameter the fitting function Eq. 31 was used to set the initial parameters of the object to produce a flexural rigidity that was close to the target value. Then the object's flexural rigidity was measured to get the actual flexural rigidity of the object with the initial parameters. The flexural rigidity with these initial parameters may differ slightly from the target value because the parameter fitting function was obtained by many approximations. This difference in value can be used to readjust the parameters to get the final object parameters which will produce a flexural rigidity value that is very close matching to the target. The initial and final parameter values of the two references are shown in table 11.

The anchors and springs that surround the object's particles provide structural strength to the object. The Tensegrity representation method provides objects with adjustable rigidity, even for objects without geometric stiffness such as a series of particles. The results of the particle-to-particle constant parameter analysis prove that the connection between particles does not have a significant effect in determining the flexural rigidity. Therefore, the elasticity-rigidity regulation of the Tensegrity representation object does not depend on the formation of the object's particles.

Table 11: Tensegrity representation object's parameter values for the reference flexural rigidity of the microtubules.
The time scale is 1:4×10-6. The anchor distance parameter and the section length parameter are multiplied by 24 nm to obtain the real-world values. Therefore, the value of each parameter is 720 nm.

Parameter	Microtubules			
	Pure		With MAPs	
	Initial value	Final value	Initial value	Final value
Reference flexural rigidity (*EI*)	3.7×10^{-24} Nm^2		16×10^{-24} Nm^2	
Reference simulation flexural rigidity (EI_{sim})	81637.42		353026.7	
Particle-to-particle constant	0.1	0.1	0.1	0.1
Anchor-to-anchor constant	0.17	0.196	0.18	0.172
Anchor-to-particle constant	0.1357	0.145	0.137	0.137
Anchor distance	30	30	30	30
Number of sections	14	14	14	14
Section length	30	30	30	30
Bridge connection	0	0	3	3
Length of the object	10.056 µm	10.056 µm	10.056 µm	10.056 µm
Simulation flexural rigidity (EI_{sim})	79805.74	81638.99	366449.437	352210.6
Flexural rigidity (*EI*)	3.617×10^{-24} Nm^2	3.7×10^{-24} Nm^2	16.61×10^{-24} Nm^2	15.96×10^{-24} Nm^2

4.6.4 Maximum spring constants

The use of springs in simulations with Euler integration needs to be carefully regulated. Since Euler integration sums up all the forces in each computation cycle, the accumulation of forces on the object's particles that are too large can cause overshoot. Setting the spring constant too high can cause an imbalance of the spring forces, especially on large time courses. If the spring forces are unbalanced the object's particles can vibrate violently. If the imbalance is too large, then overshoot occurs, and the object explodes.

There are three types of springs in the Tensegrity representation object: particle-to-particle springs, anchor-to-anchor springs, and anchor-to-particle springs. In the previous section, the particle-to-particle constant parameter was shown to have no significant effect on the flexural rigidity of objects. Therefore, we can set particle-to-particle springs aside in the discussion. On the other hand, the anchor-to-anchor constant and anchor-to-particle constant parameters have a positive linear correlation with flexural rigidity. To make the object stiffer, this spring constant parameter can be set higher. However, a spring constant that is too high can cause excessive force and overshoot. Therefore, the maximum spring constant limit must be determined.

Table 12 shows a list of anchor-to-anchor constant values and anchor-to-particle constant values as well as object reactions in the bending evaluation test. Tensegrity objects were stable at the correct spring constants. However, it started vibrating in a certain combination of the spring constants and explodes when the spring constants were too high. Although most of the object's particles vibrate when subjected to pressure from any force, they usually stabilize when the pressure is removed. In the results mentioned in Table 12, the vibration did not stop even after the stress was removed for a long time.

The tensegrity representation method provides a bridge connection parameter that will add springs to connect the surrounding anchors. The bridge connection parameter has been shown to significantly increase the rigidity of the object. Although the bridge connection parameter value setting is limited to one-third of the number of section parameter to influence the rigidity of the object, it significantly increases the rigidity without increasing the spring constants. Therefore, the bridge connection parameter is a means provided by the Tensegrity representation method to increase object rigidity without the risk of overshoot.

Table 12: The maximum value of the spring constant parameters of the tensegrity representation object.

Anchor-to-particle constant	Anchor-to-anchor constant				
	0.15	0.2	0.25	0.3	0.35
0.15	Stable	Stable	Stable	Stable	**Break**
0.2	Stable	**Vibrating**	Stable	Stable	**Break**
0.25	Stable	Stable	**Vibrating**	**Vibrating**	**Break**
0.3	Stable	Stable	**Vibrating**	**Vibrating**	**Break**
0.35	Stable	Stable	Stable	**Break**	**Break**
0.4	Stable	Stable	Stable	Stable	**Break**
0.45	Stable	Stable	Stable	Stable	**Break**
0.5	Stable	Stable	Stable	Stable	**Break**
0.55	Stable	Stable	Stable	Stable	**Break**
0.6	Stable	Stable	Stable	Stable	**Break**
0.65	Stable	Stable	Stable	Stable	**Break**
0.7	Stable	Stable	Stable	Stable	**Break**
0.75	Stable	Stable	Stable	Stable	**Break**
0.8	Stable	Stable	Stable	Stable	**Break**
0.85	Stable	Stable	Stable	Stable	**Break**
0.9	Stable	Stable	Stable	Stable	**Break**
0.95	Stable	Stable	Stable	Stable	**Break**
1	Stable	Stable	Stable	**Vibrating**	**Break**
1.5	**Break**	**Break**	**Break**	**Break**	**Break**

4.7 Performance and scalability

The Tensegrity representation method was implemented using the simulation system platform to present many microtubule objects. Each microtubule object was represented by a

length of 100 particles. Simulations of Brownian motion were applied to microtubule objects that made them moved randomly as if random hydrodynamic forces from the surrounding solvent had moved them. The number of particles in the simulation system and the number of microtubule objects were increased and the performance of the graphical framerate is observed. 90 Hz performance in dual stereo displays is expected for VR performance requirements. This system runs with the following hardware:

- Intel® Core™ i7-6850K CPU @ 3.60GHz processor.
- GPU GeForce GTX 1080.
- 16.0 GB RAM.

Table 13 shows the two performance results of the Tensegrity representation method in the system. Performance 1 was achieved by maintaining a graphics frame rate of at least 90 Hz, while performance 2 was obtained without the need for a 90 Hz frame rate. Performance 1 produced large number of simulation entities (particles, anchors, and springs) in good VR performance. Performance 2 shows more than four times the simulation entities while still in VR performance which can be excluded. This shows that the scalability of the system to provide simulation entities can be increased at the expense of graphics performance. Perhaps, in a non-VR simulation such as in the NTSC-M 30 Hz standard TV broadcast, the Tensegrity representation method can provide enormous number of simulation entities.

Table 13: Performance of the Tensegrity representation method in the simulation system.

	Performance 1	Performance 2
Number of particles	12,800	51,200
Number of anchors	5,500	22,440
Number of springs	36,500	148,920
Framerate	±90 Hz	±55 Hz

The simulation only executes the object collision process with the GPU compute shader, while the other computation processes are carried out with a multicore CPU. The reason for not applying all processes to the GPU is because the purpose of this research is to analyze and design an object creation method; therefore, some observation and analysis programming code is still required to run between processes. Optimizing performance using GPU compute shaders

and multiple GPUs is another interesting research topic in the future that is not within the scope of current research.

The performance of the Tensegrity representation method can be improved in several ways:

- Store all data in GPU memory and execute all processes using GPU compute shaders without copying data back and forth to CPU memory.
- Use more powerful hardware, especially the GPU.
- Use multiple GPUs for all the processes.

The improvements are expected to be huge, especially when deploying multiple GPUs [13][14]. Implementing this strategy requires complex specialized programming techniques which are another topic of research.

4.8 Comparison with other particle object methods

This study compares the Tensegrity representation method with three particle object methods used or potentially used in molecular simulations: the spring compound method[5], the As-Rigid-As-Possible (ARAP) method with oriented particles [52] and the Unified particle physics [53][56] . The spring compound method is a method used in the particle simulation system which is the basis of this research. This has been used in microtubule gliding assay simulation [12][13][14]. The ARAP method and the Unified particle physics method have never been used in molecular simulations, but they have the potential to be used. Table 14 summarizes some of the comparisons.

The spring compound method produces only a small degree of rigidity. The rigidity is mainly determined by the formation of particles, which are limited by the shape of the object. Objects without geometric stiffness such as a series of particles find it difficult to become rigid with this method. For long filamentous objects, this method fails to propagate any force to the entire object's particles, resulting in incorrect bending behavior of the object.

The ARAP methods can present objects with varying flexibility. However, the objects it creates are not rigid objects and too flexible to be rigid. Therefore, it can be said that this method creates elastic objects with a wide range of flexibility but with a small degree of rigidity. Another disadvantage of this method is that it does not use the GPU for physics calculations and has a performance bottle in collision detection processing. Therefore, this method is not scalable and cannot simulate large number of objects in the VR interface.

Table 14: Comparison between Tensegrity representation method and other particle object methods.

	Tensegrity representation method	Spring compound method	ARAP method	Unified particle physics method
Type of object	Elastic objects.	Elastic objects.	Deformable objects and elastic objects.	Gases, liquids, deformable solids, rigid bodies, cloth.
Time scale dependency	Rigidity depends on the time step.	Rigidity depends on the time step.	**Independent of time step.**	Rigidity depends on the time step.
Physics accuracy	**Wide range of elasticity, accurate bending shape.**	Small degree of rigidity.	For visually plausible deformation.	Plausible but not accurate.
GPU calculation	**GPU for physics.**	**GPU for physics.**	No GPU for physics.	**GPU for physics.**
Degree of flexibility	**Wide range of flexibility.**	Small degree of flexibility.	**Wide range of flexibility.**	Small degree of flexibility.
Degree of rigidity	**Wide range of rigidity.**	Almost no rigidity.	Small degree of rigidity.	**Wide range of rigidity.**
Viscoelasticity simulation	**Shows viscoelasticity.**	**Shows viscoelasticity.**	Does not consider viscosity.	Does not consider viscosity.
Overshoot risk	Overshoot possible.	Overshoot possible.	**No overshoot.**	**No overshoot.**
Performance stability	**Stable, collision does not affect performance.**	**Stable, collision does not affect performance.**	Collision is performance bottleneck.	**Stable, collision does not affect performance.**

The Unified particle physics method calculates physics using the GPU and has no performance bottlenecks in collision detection. It can present many types of objects but for visually plausible, not for accurate simulations. Object's flexibility occurs in rigid objects because this method cannot present absolute rigidity. Therefore, the elasticity of objects is difficult to control. To make objects flexible, this method uses springs. Therefore, like the compound spring method, the formation of particles affects the rigidity. For this reason, this method cannot bring rigidity to an object without geometric stiffness.

The Tensegrity representation method is the only method here that can create the elasticity

of objects in various values from very flexible to very rigid and independent of the geometric stiffness of the object. The advantages of the Tensegrity Representation method compared to other methods are it makes objects with a wide elasticity range from very flexible to very rigid and the elasticity of objects does not depend on particle formation. Other advantages of this method over several other methods are GPU for physics calculations that the ARAP method does not use; viscoelastic behavior not considered by ARAP and Unified particle physics methods; and performance stability that is still lacking in the ARAP method because of the bottle neck in collision detection processing.

The disadvantages of the Tensegrity representation method are that the time scale affects the rigidity of the object and there is still the possibility of overshoot. The time scale is the ratio between the real-time value and the time in the simulation, as shown by t_{sim} in Eq. 23. The time scale value depends on simulation performance as it decreases as performance (compute cycles per second) decreases. Therefore, if the simulation performance is stable, the time scale is stable, and the object rigidity is also stable. The Tensegrity representation method and its particle simulation system work stably and are not affected by the collision detection process; therefore, the rigidity in this method is stable. Regarding the risk of overshoot, if the spring constant has been set below the limit as shown in Table 12, the object overshoot is unlikely to occur.

4.9 More complex mechanisms

The development of the Tensegrity representation method is further enhanced by implementing several types of joints. This implementation can improve the usefulness of the object methods in molecular simulations by providing more complex mechanisms. The idea for this feature came from the method for rigid body joints in real-time Unified particle physics developed by Lovrovic and Mihajlovic [56]. They apply some joint particles as shared possession particles between two rigid bodies. The formation of the joint particles determines the type of connection between the two rigid objects. In our developed Tensegrity representation method, we use some anchors as joint anchors connecting two objects. The use of joint-anchors has the advantage of being abstract objects that cannot visible or collided; therefore, the joint-anchors can be placed in any position to be mechanically optimal without the need to insert them in the object shape.

The hinge joint is implemented by adding two joint anchors perpendicular to the connection of two objects (Fig. 25A). This joint eliminates one swing direction and eliminates rotation to

leave only one swing direction. A rotational joint also uses two joint anchors, but they are placed parallel to the connection of the two objects (Fig. 25B). This joint eliminates the two-way swing directions leaving only rotation. Ball-and-socket joint (spheroid joint) is implemented by adding one joint anchor to the joint position of the two objects, thus allowing all directions of movement and rotation of the two objects (Fig. 25C). This joint only functions as a distance constraint between the two objects. It is also possible to implement other types of joints by this concept.

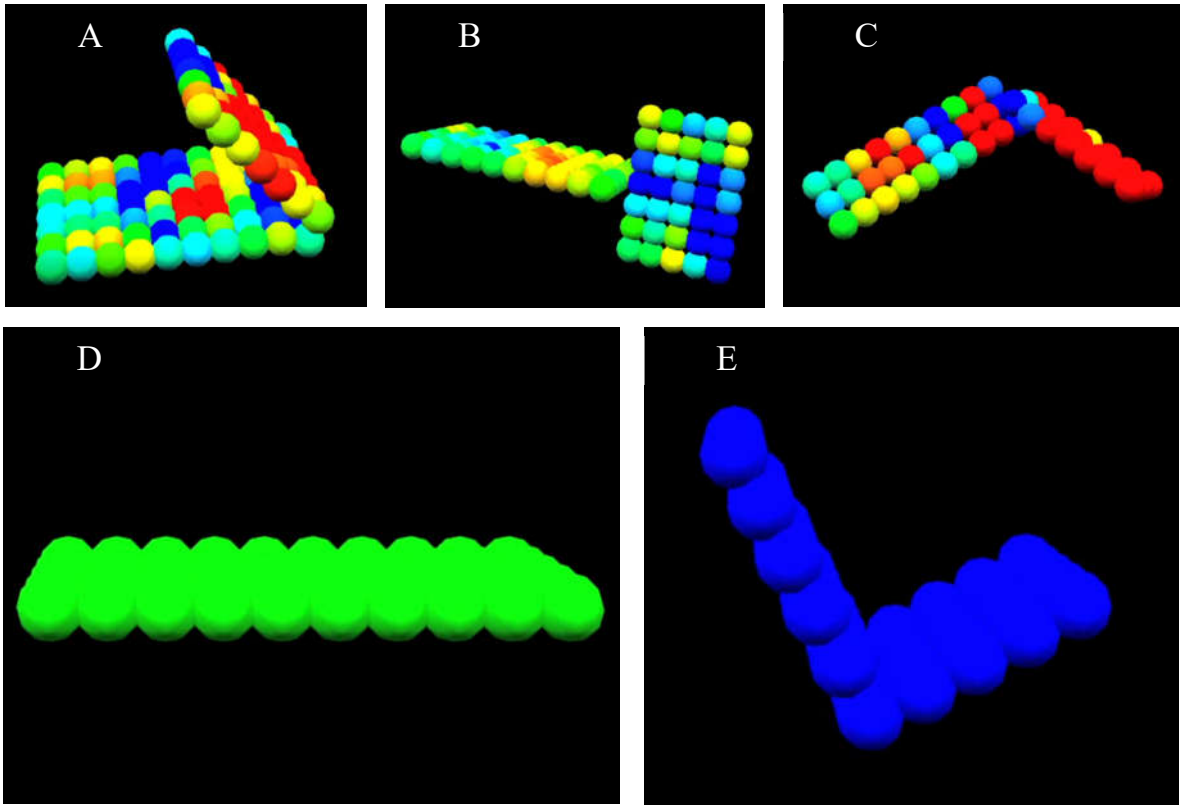


Figure 25: Implementation of a more complex mechanism in the Tensegrity representation method.
(A) Hinge joint. (B) Rotational joint. (C) Ball-and-socket joint (spheroid joint). (D) The rectangular object before shape transforming into (E) an L-shaped object.

Additional mechanism capability in the Tensegrity representation method is implementing direct shape transformation. An object can suddenly change shape after a trigger is activated. For example, a rectangular object suddenly folds into an L-shaped object by pressing a key on the keyboard and unfolds after pressing the key one more time (Fig. 25D and 25E). This process is done by overwriting all object spring distance data with other data from one of the two hidden objects. These two hidden objects are identical objects with folded and unfolded

shapes. These shape transformations may be useful in molecular simulations for transformations such as cis-trans isomer transformation.

4.10 Achievements

The Tensegrity representation method was developed to create static elastic model of biomolecular 3D objects to be deployed in an interactive haptic VR simulation platform. This chapter makes the following achievements:

- **A static force-balancing model.**

  The Tensegrity representation method has created a force-balancing springs structural framework for 3D objects. A static flexural rigidity measurement was done by bending the object to find the maximum deflection. This measurement process also allows the deflection of each object's particle to be analyzed using the classical bending equation and the results confirm the bending shape of the object. By analyzing the retardation and relaxation processes in this measurement process, the viscoelastic behavior of objects can be observed, and the results are in accordance with the Kelvin (Voigt) rheological model.

- **Object's parameters fitting function for flexural rigidity.**

  The correlation between the object's parameters and the flexural rigidity of the object has been analyzed to find the parameter fitting function. This function is useful for getting the object's flexural rigidity that is close to the desired value by adjusting its object parameters. Resetting the anchor-to-anchor constant parameter or the anchor-to-particle constant parameter can be done for a more precise result with the desired value.

- **Objects with wide range of elasticity with fine resolution from very flexible to very rigid objects.**

  Object's parameters in the Tensegrity representation method can be adjusted to match the desired flexural rigidity value. The wide range of values that can be applied to the parameters allows the object's flexural rigidity to be set in a wide range of values from very flexible to very rigid. Two of these parameters have a positive linear correlation with the flexural rigidity, which makes the flexural rigidity adjustable in fine detail values.

- **Objects with independent elasticity/rigidity from the particle formation.**

  The Tensegrity representation method creates a structural framework of springs to support the object conformation. By using this technique, the flexural rigidity of the object does not depend on the formation of the particles of the object. Therefore, objects without geometric rigidity can have rigidity if they are created as Tensegrity representation objects.

4.11 Significances

The Tensegrity representation method can create particle objects with a wide range of elasticity regardless of the particle structure. This solves the limitations of the existing particle object method of providing rigidity to objects without geometric stiffness. As a result, biomolecular objects can be created with the smallest possible number of particles while preserving their mechanical properties. With the minimum number of particles needed to form biomolecular objects, the number of biomolecular objects that can be handled by the simulation system is very large. Therefore, simulations of natural molecular phenomena can be carried out using large-scale biomolecular 3D objects.

4.12 Problems and limitations

In developing the Tensegrity representation method, some problems were found. This problem is the limitation of this Tensegrity representation method. Overcoming these limitations is beyond the scope of this study and will be an interesting challenge in the future. These limitations are explained as follows:

- **The Tensegrity representation method cannot create anisotropic objects.**
  Due to the need to efficiently use the number of particles in the object, the simplest particle object representation is used for biomolecular 3D objects. The details of the structure of biomolecules are neglected in this coarse-grained degree. Therefore, the anisotropic property of the object is ignored. For example, microtubules are hollow tubular objects whose flexural rigidity varies according to the degree of bending due to cross-sectional flattening [44]. However, because a microtubule is represented as a series of particles, there is no cross-sectional flattening in the object. This is not a significant issue for the time being because most experimental work also considers microtubules to be isotropic.
- **The Tensegrity representation method cannot create objects with absolute rigidity.**
  Spring is the main element used to create the structural strength of the Tensegrity representation object. Due to their nature, springs always present elasticity. Therefore, any object made with springs will always exhibit elasticity and never become completely rigid. The risk of object overshoot also prevents the Tensegrity representation object from increasing its spring constant to make the object stiffer. The Tensegrity representation method has provided a bridge connection parameter to increase the object rigidity, which allows the object to be very rigid even though it is not completely rigid. The fact that

biomolecular objects are soft and tend to be elastic makes objects with absolute rigidity are rarely needed in molecular simulations.

Chapter 5

Discussion

This chapter discusses the implementation of Tensegrity representation objects on the interactive haptic VR simulation platform in simulating natural molecular phenomena. Interactive parameters live control is also discussed to reproduce simulations of experimental phenomena that evaluation functions are too difficult to define using existing theories. We also compared the interactive haptic VR simulation platform developed in this study with other VR molecular systems. Finally, the main contributions of this thesis research are outlined.

5.1 Coarse-grained trade-off

The interactive haptic VR simulation platform developed in this study is intended to run large-scale biomolecular simulations in a VR environment. The need to simulate large numbers of objects and a stereo graphics performance of at least 90 Hz does not only require a powerful simulation engine but also requires an efficient object model. The only way to achieve this demand is to use a coarse-grained object model. The lightness of coarse-grained computations is obtained at the expense of the precision and accuracy of simulation results because some object details are neglected. This is where the trade-off between the performance and accuracy of the coarse-grained model occurs.

The interactive haptic VR simulation platform is a tool whereas the simulation of natural molecular phenomena is the product. Each phenomenon has a different implementation. The need for precision and accuracy affects the degree of coarse-grained. Therefore, the simulation features that are sacrificed differ from one another depending on the simulation being developed.

The trade-off in a coarse-grained simulation is between time (time course), space (dimension / size / mass), and force (external force). Time is a predetermined time course (time step) to achieve the simulation duration of the presented phenomenon. The value of time course varies according to the phenomenon. Some phenomena take time course in microseconds, milliseconds, seconds, even minutes. Space is the space scale that determines the size and mass of the molecular objects in the simulation. A finer detail simulation requires the particles to represent smaller objects, hence it requires more particles in the simulation. This will affect the

scale of the simulated phenomena and the simulation performance as well. A force is any force that exists in a simulation which is usually influenced by predetermined time and space. Apart from these quantities, the elasticity / rigidity and viscosity of the object are also quantities that affect the visual presentation of the simulation. Therefore, object rigidity and viscosity are additional features that can be sacrificed / manipulated to reproduce the phenomena in the simulation. Figure 26 illustrates the trade-off.

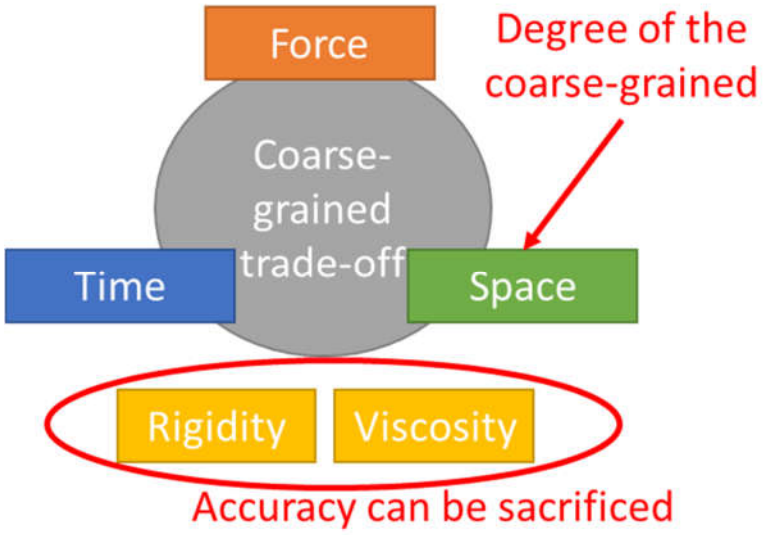


Figure 26: Coarse-grained simulation trade-off.
The trade-off is between time (time course), space (size and mass), and force, while the object rigidity and the viscosity are additional features that can be sacrificed.

The space scale, which determines the size and mass of each particle, is usually the first quantity to be determined. This determines the degree of the object's coarse-grained. The purpose of this setup is to adjust the simulation to the detail and volume needed to reproduce the natural phenomenon. The second quantity that is usually determined is the time course, because it is related to the duration of the simulation needed to present the natural phenomenon. The length of each phenomenon is different from one another. After the two quantities are determined, the forces that occur in the simulation will follow. The actual value of the force can be determined based on time, size, and mass. Due to the limitations of Euler integration, the visual behavior of molecular objects may not be accurate. The Euler integration works best in the shortest possible time course. Because a larger time course requires bigger force, the accuracy of the simulation may reduce. Applying too much force in the particle simulation can also make the object's shape (bending) change incorrectly (Fig.27). To fix the object behavior,

the rigidity and viscosity can be optimized at the runtime. By increasing the object rigidity, deformation can be reduced. Interpretation by specialists is necessary to reproduce the correct phenomenon.

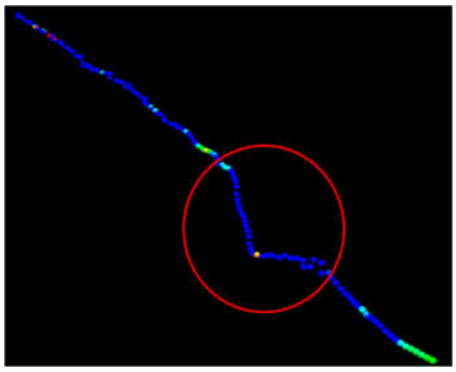


Figure 27: Incorrect bending of a filament object when the force is too large. Increasing the rigidity of objects can reduce defects for better visual results.

5.2 Interactive optimization

In a massive swarm of biomolecular objects, some interesting global dynamics may emerge. However, most of the global dynamics are temporal in nature, making it difficult to determine the evaluation function for their emergence. Meta-search algorithms such as genetic algorithms are difficult to apply to find the optimal combination of parameters that govern this global dynamic, due to the lack of evaluation functions. The most likely way to find this optimal set of parameters is to adjust them at runtime to control the parameters to reproduce this global dynamic behavior [5].

Interactive parameter optimization is part of the interactive VR haptic simulation platform that we developed. The purpose of this feature is to visually optimize simulation parameters at runtime to reproduce natural phenomena that the fitting function is too difficult to obtain. Parameters that can be optimized at runtime are simulation environment parameters, object parameters, and existing forces. Simulation environment parameters are parameters that have a global effect on all particles in the simulation, such as drag / dampening, gravitational force, and collision parameters. Object parameters are all parameters of the Tensegrity representation method, most of which are optimized to adjust object rigidity.

The rigidity of objects affects the objects dynamics movement in a massive swarm. This affects the pattern of the emergence of global dynamics by changing the size of the circular motion of curved objects. For example, in the microtubule gliding assay simulation

[12][13][14], some microtubules make circular motions that are too small compared to the actual experimental results. This happens because the spring compound method used cannot stiffen the microtubule object; therefore, they are too easy to bend. More rigid microtubule objects are needed to improve simulation results that are closer to experimental results.

Most of the Tensegrity representation object parameters affect the object rigidity except for the particle-to-particle constant parameter. However, the anchor-to-anchor constant parameter and the anchor-to-particle constant parameter are preferred to be used in the interactive parameter optimization because they have a smooth positive linear correlation with the object flexural rigidity. Parameter adjustment is done by overwriting object parameter constants with new values that increase or decrease according to the user's decision after observation. The user decides on the optimization by interactive visual observation at runtime. Optimization needs to be performed by users who specialize in simulated experimental phenomena.

The Tensegrity representation method provides a new feature in the interactive haptic VR simulation platform we developed. It provides the ability to manipulate object rigidity which provides new options for interactive parameter optimization. The role of the specialist user will determine the reproduction results of the simulated experimental phenomena. The synergy between human intuition and machine computation efficiency results in the ability to optimize simulation parameters that the existing meta-search algorithms cannot perform due to a lack of evaluation functions.

5.3 Comparison with other VR molecular systems

This study compares an interactive haptic VR simulation platform we develop with other existing VR molecular simulation systems: Molecular Rift [29], 3D-Lab [30], Caffeine [57], Molecular Dynamic Visualization (MDV) [58], VR models of breast cancer cells [28], ChimeraX, AltPDB, MolecularZoo [31], and iMD-VR [59]. These comparisons are summarized in Table 14.

Most of the existing VR molecular systems are a kind of molecular visualization system that does not simulate physics or provides no user interaction with molecular objects. From the list of other VR molecular systems in Table 15, only MolecularZoo and iMD-VR use physics simulation. However, both VR systems do not present deformable / elastic objects. The reason could be due to performance. Deformable / elastic object simulations are very computationally expensive while simulating rigid objects is significantly cheaper in computation. The number of objects that can be handled in the simulation is also limited for both VR systems. This is

because they do not use the GPU for physics simulation. The application of GPU computation for physics simulation is still rarely done because the complexity of physics simulation makes it difficult to parallelize. Therefore, physics simulation using GPU computing in the interactive haptic VR simulation platform that we developed is a major advantage over all the other VR molecular systems.

Another aspect that needs to be considered in simulation performance is the stability of performance in various events. Some simulation systems can degrade their performance during heavy and uncertain events such as collisions. Collision detection and calculation of physical responses are usually complex and take a long computation time. Because collisions are not regular events, their erratic appearance can result in volatility or instability in performance. This happened at MolecularZoo which reportedly had bottleneck performance in a collision detection event. Unlike MolecularZoo, the interactive haptic VR simulation platform we have developed properly parallels all physics events and treats each particle collision event as computationally routine. Therefore, the performance of this VR simulation platform is stable without a bottleneck in any physics event.

In terms of user interface, the interactive haptic VR simulation platform provides a natural user interface in the form of virtual hands. The virtual hands can even interact with 3D objects as if they were in a simulated world. Haptic feedback provides a more intuitive interaction for sensing the dynamics of molecular objects. The hand user interfaces are also provided at Molecular Rift, 3D-Lab, and MolecularZoo; however, this is only for navigation. Caffeine and Molecular Dynamics Visualization do not provide a hand interface, whereas other VR systems do provide a controller stick for navigation. This haptic hand user interface is also a great advantage of the VR simulation platform we have developed.

The only disadvantage of the VR simulation platform compared to some other VR molecular systems is the absence of VR collaboration feature. 3D-Lab, Molecular Dynamics Visualization, ChimeraX, and AltPDB provide collaboration feature with other users in the same simulation scene. This feature usually displays each user as an avatar for other users to see and interact with. This is an interesting feature that can be added to our platform and we are working on it.

Table 15: Comparison between the Interactive haptic VR simulation platform we have developed with other VR molecular systems.

	User interface	Physics simulation	Viscoelastic simulation	Number of objects	Performance stability	Haptic feedback	Collaboration VR
Our VR simulation platform	**Hand user interface with physics**	**Physics for objects and hands with GPU computing**	**Simulates elasticity and viscosity**	**Presents many objects**	**Stable (independent from collision events)**	**Supports custom haptic device**	Not yet, but possible in the future
Molecular Rift	Hand user interface for navigation only	No physics but can move objects	No physics simulation	**Presents many objects**	**Stable (no physics, no collision)**	No haptic feedback	Not provided
3D-Lab	Hand user interface for navigation only	No physics but can move objects	No physics simulation	**Presents many objects**	**Stable (no physics, no collision)**	No haptic feedback	**Yes**
Caffeine	No hand interface	No physics simulation	No physics simulation	**Presents many objects**	**Stable (no physics, no collision)**	No haptic feedback	Not provided
MDV	No hand interface	No physics simulation	No physics simulation	**Presents many objects**	**Stable (no physics, no collision)**	No haptic feedback	**Yes**
VR model of a breast cancer cell	Using controller sticks for navigation only	No physics simulation	No physics simulation	**Presents many objects**	**Stable (no physics, no collision)**	No haptic feedback	Not provided
ChimeraX and AltPDB	Using controller sticks for navigation	No physics but can move objects	No physics simulation	Limited number of objects	**Stable (no physics, no collision)**	No haptic feedback	**Yes**
MolecularZoo	Hand user interface for navigation only	Provides collision between objects, but not with hands	No deformable object	Limited number of objects	Collisions reduce the performance	No haptic feedback	Not provided
iMD-VR	Using controller sticks for navigation	Can move objects & provides interaction between objects	No deformable object	No data	No data, possibly stable	No haptic feedback	Not provided

5.4 Contributions

This study has produced an interactive haptic VR simulation platform along with a novel object creation method to simulate natural molecular phenomena. The development of a static elastic object model for large-scale biomolecules in an interactive haptic VR simulation platform has contributed to several achievements.

- Implementation of VR molecular system with haptic feedback with:
    - physics simulation for large-scale of biomolecular objects,
    - natural hand user interface with haptic feedback postulating the tactile sense of touching biomolecular objects,
    - interactive parameter optimization to adjust the elasticity / rigidity of objects at runtime to reproduce natural phenomena.
- The tensegrity representation method, a novel unified particle object method for creating molecular 3D objects with features:
    - wide range of object elasticity with fine resolution from very flexible to very rigid objects,
    - elasticity which is independent of the formation of the object's particles. This feature allows the Tensegrity representation method to create and provide rigidity to objects without geometric stiffness.
    - more complex mechanisms such as hinge joint, rotational joint, ball-and-socket joint, and shape transformation.

The interactive haptic VR simulation platform contributes to providing a tool to simulate natural phenomena with large-scale biomolecules in a VR environment with haptic feedback. The Tensegrity representation method contributes to providing an alternative solution for creating flexible 3D objects in a particle simulation system that solves the problem of presenting a wide range of object elasticity as well as rigidity in objects without geometric stiffness.

Chapter 6

Conclusions

This thesis has developed a Tensegrity representation method to model static elastic objects for large-scale biomolecules in an interactive haptic VR simulation platform. The VR simulation platform is implemented as a particle simulation system capable of GPU computing to enable large-scale biomolecular simulations in VR performance. A haptic rendering concept to render tactile feelings on biomolecules has been postulated and applied via custom-built haptic rendering device. The Tensegrity representation method has succeeded in creating 3D objects with a wide range of elasticity, from very flexible objects to very rigid objects. The elasticity of the tensegrity representation object also does not depend on the formation of the object's particles, allowing objects without geometric stiffness to have rigidity. The Tensegrity representation method also makes it possible to create more complex mechanisms such as hinge joint, rotational joint, ball-and-socket joint, and shape transformation. This makes the Tensegrity representation method a method of making potential objects for biomolecular simulations.

6.1 Future works

The work in this thesis is promising to become a platform for creating haptic VR simulations of large-scale biomolecules to preview some experimental results to aid decision making for experimentalists. However, this work is still in its early stage. Several aspects can be addressed for future works to improve accuracy to make the simulation more realistic.

- **Dynamics elasticity measurement.**
  The flexural rigidity measurement carried out in this thesis was a static elasticity measurement method which measures the flexural rigidity of an object from the maximum deflection of the endpoint of the object in a static state. In microtubule flexural rigidity measurement experiments, the static elasticity measurement is more consistent in finding the flexural rigidity value because of its direct measurement on the material [19], whereas dynamic elasticity measurement often yields different results by up to two orders of magnitude [45]. The difference in results may be due to the involvement of uncontrolled forces such as hydrodynamic forces. The hydrodynamic

forces change continuously both along the filament object and over time [19]. Another hypothesis is that the various measured values are due to the flattening of the microtubule body which makes the degree of bending affect the value [44].

Despite the problems, dynamic elasticity measurements show dynamic behavior and properties that are not shown by static measurements. Viscosity and anisotropic behavior are related to this. Simulating dynamic elasticity measurements is indeed a very challenging work in the future. Establishing springs and anchors to meet these challenges is very complex and could be an interesting topic for future research.

Another challenge related to object viscosity is that high viscosity will reduce the object's mobility. However, some molecular objects do require high viscosities. To solve this problem, one possible solution is to apply a damper to each spring on the Tensegrity representation object to allow the viscosity can be set individually for each object.

- **Effective fluctuation reduction for large time step simulations.**

  The Euler integration calculates the force of each compute cycle in a defined time step. In large time steps, some detail of what happened between them may be lost. The larger the time step, the greater the force calculated for each compute cycle. When a force is applied to a biomolecular filament object, too much force can cause erratic motion which makes the object fluctuate and visually appear to break. To reduce this defect, the object rigidity can be adjusted more rigidly so that the broken part is less visible. However, that would not completely remove the flaw. Effective fluctuation reduction is required when simulating a large time step.

  Reducing fluctuations is not a simple task as Euler integration problems have less accurate results in a larger time step. The solution is not always in the object method but can also be in the simulation system by switching to a more sophisticated integration method, e.g. Runge-Kutta, implicit integration, and adaptive time steps. However, this method is more complicated which can hurt performance. Whether they can be implemented by GPU computing or not remains a question that needs to be answered in future work. Calculating the force differently is also an alternative. Adjusting the force duration and the area of impact can be a potential solution between performance and plausible dynamic appearance.

6.2 Limitations

Tensegrity representation method and interactive VR haptic simulation platform are combined into a solution in this thesis. It has several limitations that its implementation should consider the

following points.

- **It cannot be used when the anisotropic behavior of the object is an important property to simulate.**

  The Tensegrity representation method was developed to create objects with a minimum number of particles to achieve large-scale biomolecular simulations in VR performance. With this degree of coarse-grained, some details of the object's properties are neglected. Anisotropic objects are not considered because other experiments usually also consider all objects to be isotropic. Presenting anisotropic properties into an object is a complicated task which cannot be displayed using the current Tensegrity representation method. Therefore, if it is important to simulate objects with anisotropic behavior, then it is better not to use the VR simulation platform that we developed in this thesis.
- **It cannot be used to simulate absolute rigid objects over large time course if high accuracy and precision are demanded.**

  The Tensegrity representation object is formed by particles supported by springs and anchors. Due to the nature of the spring which reacts after a change in distance, the Tensegrity representation object is never absolutely rigid. The risk of overshoot that must be avoided also prevents the object's spring constant parameters from being set very high. Therefore, the Tensegrity representation method cannot be claimed to make objects rigid, even though it can create very rigid objects. If high accuracy and precision are required in simulating absolute rigid objects over a large time course, this VR simulation platform is not a suitable tool to use.
- **It cannot replace the results of laboratory experiments but is used to preview possibilities in experiments and help for planning and decision making.**

  This simulation system platform uses a coarse-grained model that ignores some of the detailed properties in natural phenomena. The accuracy and precision of the results do not precisely match the actual phenomenon. Some inaccuracies may occur and accumulate during computation. Therefore, the results from this simulation platform are not valid enough to replace real wet experiments. However, this simulation platform can be used to develop simulations in the time course and duration required for natural phenomena, something molecular dynamics simulations — which are believed to be accurate — cannot achieve. Additional 3D VR views, a hand user interface with sensible haptic feedback, and interactive parameter optimization, provide a more intuitive way to observe and provide a 3D viewing angle that traditional laboratory experiments

cannot provide. Therefore, this interactive haptic VR simulation platform can be used to preview possibilities in experiments and aid planning and decision making, as long as it is not claimed to replace actual laboratory experiments.

Acknowledgment

All praises to the Almighty Allah, the Most Gracious and the Most Merciful, for the strengths and His blessing in completing my doctoral degree.

I wish to show my gratitude to my supervisor, Masayuki Yamamura sensei, for his precious guidance and support throughout my thesis with his patience and expertise. Deepest gratitude is also due to Akihiko Konagaya sensei, my former supervisor, for his dedicated supervision and constant encouragement towards the completion of my study. Special thanks to Gregory Spencer Gutmann for his sharing knowledge, technology, and invaluable assistance.

I have also benefited greatly from knowing and working with Yutaka Ueno sensei from National Institute of Advanced Industrial Science and Technology, Satoru Tsutoh from Fuji Xerox Co., Ltd, Akira Kakugo sensei from Hokkaido University, and all the member of the Artificial Molecular Muscle project of New Energy and Industrial Technology Development Organization (NEDO) Japan.

I would also like to thank all the member of Konagaya Laboratory, especially Bulbul Mahmot who became my tutor and helped me settle down during my arrival in Japan. Not to forget all fellow students from Indonesia and Muslim students at the Tokyo Institute of Technology.

I owe my loving thanks to my supportive, encouraging and patient wife Rahmawati Mahendra Dewi. Also, thanks to my sons Ahnaf Arifian Adisubroto and Faris Ahnafi Adisubroto. Without their encouragement and understanding, I would never finish my study. My special gratitude goes to my beloved mother Siti Maryam, my father-in-law Tedjo Yuwono, my late father Dalil Adisubroto for his spiritual strength, and all family members for being very supportive in spiritual and understanding during my study.

Finally, I would like to thank the Indonesia Endowment Fund for Education (LPDP) for supporting me with a scholarship.

References

[1] D.E. Ingber, Tensegrity I. Cell structure and hierarchical systems biology, J. Cell Sci. 116 (2003) 1157–1173. https://doi.org/10.1242/jcs.00359.

[2] R.L. Swanson, Biotensegrity: A unifying theory of biological architecture with applications to osteopathic practice, education, and research-a review and analysis, J. Am. Osteopath. Assoc. 113 (2013) 34–52. https://doi.org/10.7556/jaoa.2013.113.1.34.

[3] D. Inoue, G. Gutmann, T. Nitta, A.M.R. Kabir, A. Konagaya, K. Tokuraku, K. Sada, H. Hess, A. Kakugo, Adaptation of Patterns of Motile Filaments under Dynamic Boundary Conditions, ACS Nano. 13 (2019) 12452–12460. https://doi.org/10.1021/acsnano.9b01450.

[4] K. Matsuda, A.M.R. Kabir, N. Akamatsu, A. Saito, S. Ishikawa, T. Matsuyama, O. Ditzer, M.S. Islam, Y. Ohya, K. Sada, A. Konagaya, A. Kuzuya, A. Kakugo, Artificial Smooth Muscle Model Composed of Hierarchically Ordered Microtubule Asters Mediated by DNA Origami Nanostructures, Nano Lett. 19 (2019) 3933–3938. https://doi.org/10.1021/acs.nanolett.9b01201.

[5] G. Gutmann, R. Azuma, A. Konagaya, A Virtual Reality Computational Platform Dedicated for the Emergence of Global Dynamics in a Massive Swarm of Objects, J. Imaging Soc. Japan. 57 (2018) 647–653. https://doi.org/10.11370/isj.57.647.

[6] S. Stam, S.L. Freedman, S. Banerjee, K.L. Weirich, A.R. Dinner, M.L. Gardel, Filament rigidity and connectivity tune the deformation modes of active biopolymer networks, Proc. Natl. Acad. Sci. U. S. A. 114 (2017) E10037–E10045. https://doi.org/10.1073/pnas.1708625114.

[7] R. Azuma, S. Kishi, G. Gutmann, A. Konagaya, All-atom molecular dynamics of film supported flat-shaped DNA origami in water, Chem-Bio Informatics J. 18 (2018) 96–118. https://doi.org/10.1273/cbij.18.96.

[8] F. Weichert, D. Bachmann, B. Rudak, D. Fisseler, Analysis of the accuracy and robustness of the Leap Motion Controller, Sensors (Switzerland). 13 (2013) 6380–6393. https://doi.org/10.3390/s130506380.

[9] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, J. Sodnik, An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking, Sensors (Switzerland). 14 (2014) 3702–3720. https://doi.org/10.3390/s140203702.

[10] A. Pramudwiatmoko, S. Tsutoh, G. Gutmann, Y. Ueno, A. Konagaya, Haptic Rendering Applied to Hand Tracking 3D User Interface for a Molecular Modeling Environment, in: 24th Int. Symp. Artif. Life Robot., International Society of Artificial Life and Robotics, Beppu, Oita, Japan, 2019: pp. 109–114.

[11] A. Pramudwiatmoko, S. Tsutoh, G. Gutmann, Y. Ueno, A. Konagaya, A high - performance haptic rendering system for virtual reality molecular modeling, Artif. Life Robot. 24 (2019) 542–549. https://doi.org/10.1007/s10015-019-00555-9.

[12] G. Gutmann, D. Inoue, A. Kakugo, A. Konagaya, Real-time 3D microtubule gliding simulation accelerated by GPU computing, Int. J. Autom. Comput. 13 (2016) 108–116. https://doi.org/10.1007/s11633-015-0947-1.

[13] G. Gutmann, D. Inoue, A. Kakugo, A. Konagaya, Using a master and slave approach for GPGPU computing to achieve optimal scaling in a 3D real-time simulation, 2016 IEEE 11th Annu. Int. Conf. Nano/Micro Eng. Mol. Syst. NEMS 2016. 6 (2016) 95–100. https://doi.org/10.1109/NEMS.2016.7758208.

[14] G. Gutmann, D. Inoue, A. Kakugo, A. Konagaya, Parallel Interaction Detection Algorithms for a Particle-based Live Controlled Real-time Microtubule Gliding Simulation System Accelerated by GPGPU, New Gener. Comput. 35 (2017) 157–180.

https://doi.org/10.1007/s00354-017-0011-5.
[15] A. Pramudwiatmoko, G. Gutmann, Y. Ueno, A. Kakugo, M. Yamamura, A. Konagaya, Tensegrity representation of microtubule objects using unified particle objects and springs, Chem-Bio Informatics J. 20 (2020) 19–43. https://doi.org/10.1273/cbij.20.19.
[16] M. Ledbetter, Porter KR, A "microtubule" in plant cell fine structure, J. Cell Biol. 19 (1963) 239–250. https://doi.org/10.1083/jcb.19.1.239.
[17] M. Chalfie, J.N. Thomson, Organization of neuronal microtubules in the nematode Caenorhabditis elegans, J. Cell Biol. 82 (1979) 278–289. https://doi.org/10.1083/jcb.82.1.278.
[18] T. Hawkins, M. Mirigian, M. Selcuk Yasar, J.L. Ross, Mechanics of microtubules, J. Biomech. 43 (2010) 23–30. https://doi.org/10.1016/j.jbiomech.2009.09.005.
[19] M. Kikumoto, M. Kurachi, V. Tosa, H. Tashiro, Flexural rigidity of individual microtubules measured by a buckling force with optical traps, Biophys. J. 90 (2006) 1687–1696. https://doi.org/10.1529/biophysj.104.055483.
[20] H. Ghodsi, K. Darvish, Investigation of mechanisms of viscoelastic behavior of collagen molecule, J. Mech. Behav. Biomed. Mater. 51 (2015) 194–204. https://doi.org/10.1016/j.jmbbm.2015.07.015.
[21] H. Ghodsi, K. Darvish, Characterization of the viscoelastic behavior of a simplified collagen micro-fibril based on molecular dynamics simulations, J. Mech. Behav. Biomed. Mater. 63 (2016) 26–34. https://doi.org/10.1016/j.jmbbm.2016.06.006.
[22] R.B. Fuller, Tensegrity, Portf. Art News Annu. 4 (1961) 112–127.
[23] R.B. Fuller, Tensile-integrity structures, US patent 3,063,521, 1962.
[24] V. Gómez Jáuregui, Controversial origins of tensegrity, Proc. Int. Assoc. Shell Spat. Struct. Symp. (2009) 1642–1652.
[25] B.D. Matthews, D.R. Overby, F.J. Alenghat, J. Karavitis, Y. Numaguchi, P.G. Allen, D.E. Ingber, Mechanical properties of individual focal adhesions probed with a magnetic microneedle, Biochem. Biophys. Res. Commun. 313 (2004) 758–764. https://doi.org/10.1016/j.bbrc.2003.12.005.
[26] S. Hu, J. Chen, N. Wang, Cell spreading controls balance of prestress by microtubules and extracellular matrix, Front. Biosci. 9 (2004) 2177–2182. https://doi.org/10.2741/1352.
[27] C.P. Brangwynne, F.C. MacKintosh, S. Kumar, N.A. Geisse, J. Talbot, L. Mahadevan, K.K. Parker, D.E. Ingber, D.A. Weitz, Microtubules can bear enhanced compressive loads in living cells because of lateral reinforcement, J. Cell Biol. 173 (2006) 733–741. https://doi.org/10.1083/jcb.200601060.
[28] A.P.R. Johnston, J. Rae, N. Ariotti, B. Bailey, A. Lilja, R. Webb, C. Ferguson, S. Maher, T.P. Davis, R.I. Webb, J. McGhee, R.G. Parton, Journey to the centre of the cell: Virtual reality immersion into scientific data, Traffic. 19 (2018) 105–110. https://doi.org/10.1111/tra.12538.
[29] M. Norrby, C. Grebner, J. Eriksson, J. Boström, Molecular Rift: Virtual Reality for Drug Designers, J. Chem. Inf. Model. 55 (2015) 2475–2484. https://doi.org/10.1021/acs.jcim.5b00544.
[30] C. Grebner, M. Norrby, J. Enström, I. Nilsson, A. Hogner, J. Henriksson, J. Westin, F. Faramarzi, P. Werner, J. Boström, 3D-Lab: a collaborative web-based platform for molecular modeling, Future Med. Chem. 8 (2016) 1739–1752. https://doi.org/10.4155/fmc-2016-0081.
[31] T.D. Goddard, A.A. Brilliant, T.L. Skillman, S. Vergenz, J. Tyrwhitt-Drake, E.C. Meng, T.E. Ferrin, Molecular Visualization on the Holodeck, J. Mol. Biol. 430 (2018) 3982–3996. https://doi.org/10.1016/j.jmb.2018.06.040.
[32] K. Khoshelham, S.O. Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications, Sensors. 12 (2012) 1437–1454. https://doi.org/10.3390/s120201437.
[33] M.B. Stocks, S. Hayward, S.D. Laycock, Interacting with the biomolecular solvent accessible

surface via a haptic feedback device, BMC Struct. Biol. 9 (2009) 1–7. https://doi.org/10.1186/1472-6807-9-69.
[34] S.D. Laycock, M.B. Stocks, S. Hayward, Navigation and exploration of large data-sets using a haptic feedback device, ACM SIGGRAPH 2010 Posters, SIGGRAPH '10. (2010) 4503. https://doi.org/10.1145/1836845.1837007.
[35] M.B. Stocks, S.D. Laycock, S. Hayward, Applying forces to elastic network models of large biomolecules using a haptic feedback device, J. Comput. Aided. Mol. Des. 25 (2011) 203–211. https://doi.org/10.1007/s10822-010-9410-0.
[36] G. Iakovou, S. Hayward, S. Laycock, A real-time proximity querying algorithm for haptic-based molecular docking, Faraday Discuss. 169 (2014) 359–377. https://doi.org/10.1039/c3fd00123g.
[37] G. Iakovou, S. Hayward, S.D. Laycock, Adaptive GPU-accelerated force calculation for interactive rigid molecular docking using haptics, J. Mol. Graph. Model. 61 (2015) 1–12. https://doi.org/10.1016/j.jmgm.2015.06.003.
[38] G. Iakovou, S. Laycock, S. Hayward, Determination of locked interfaces in biomolecular complexes using Haptimol_RD, Biophys. Physicobiology. 13 (2016) 97–103.
[39] G. Iakovou, S. Hayward, S.D. Laycock, Virtual Environment for Studying the Docking Interactions of Rigid Biomolecules with Haptics, J. Chem. Inf. Model. 57 (2017) 1142–1152. https://doi.org/10.1021/acs.jcim.7b00051.
[40] X. Hou, O. Sourina, Six degree-of-freedom haptic rendering for biomolecular docking, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6670 LNCS (2011) 98–117. https://doi.org/10.1007/978-3-642-22336-5_6.
[41] Go Touch VR: Touch the Virtual Reality with VR Touch, (n.d.). https://www.gotouchvr.com/ (accessed September 4, 2018).
[42] I. Choi, H. Culbertson, M.R. Miller, A. Olwal, S. Follmer, Grabity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality, Proc. 30th Annu. ACM Symp. User Interface Softw. Technol. - UIST '17. (2017) 119–130. https://doi.org/10.1145/3126594.3126599.
[43] A. Desai, T.J. Mitchison, Microtubule polymerization dynamics, Annu. Rev. Cell Dev. Biol. 13 (1997) 83–117. https://doi.org/10.1146/annurev.cellbio.13.1.83.
[44] E. Memet, F. Hilitski, M.A. Morris, W.J. Schwenger, Z. Dogic, L. Mahadevan, Microtubules soften due to cross-sectional flattening, Elife. 7 (2018) 1–28. https://doi.org/10.7554/eLife.34695.
[45] H. Felgner, R. Frank, M. Schliwa, Flexural rigidity of microtubules measured with the use of optical tweezers, J. Cell Sci. 109 (1996) 509–516.
[46] M. Kurachi, M. Hoshi, H. Tashiro, Buckling of a single microtubule by optical trapping forces: Direct measurement of microtubule rigidity, Cell Motil. Cytoskelet. 30 (1995) 221–228. https://doi.org/doi:10.1002/cm.970300306.
[47] J. Van Mameren, K.C. Vermeulen, F. Gittes, C.F. Schmidt, Leveraging single protein polymers to measure flexural rigidity, J. Phys. Chem. B. 113 (2009) 3837–3844. https://doi.org/10.1021/jp808328a.
[48] K.D. Snelson, Continuous tension, discontinuous compression structures, US patent 3,169,611, 1965.
[49] R. Connelly, A. Back, Mathematics and Tensegrity: Group and representation theory make it possible to form a complete catalogue of "strut-cable" constructions with prescribed symmetries, Am. Sci. 86 (1998) 142–151. http://www.jstor.org/stable/27856980.
[50] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, R. V Kenyon, J.C. Hart, The CAVE: audio visual experience automatic virtual environment, Commun. ACM. 35 (1992) 64–72. https://doi.org/10.1145/129888.129892.

[51] T. Waltemate, D. Gall, D. Roth, M. Botsch, M.E. Latoschik, The impact of avatar personalization and immersion on virtual body ownership, presence, and emotional response, IEEE Trans. Vis. Comput. Graph. 24 (2018) 1643–1652. https://doi.org/10.1109/TVCG.2018.2794629.
[52] M.G. Choi, J. Lee, As-rigid-as-possible solid simulation with oriented particles, Comput. Graph. 70 (2018) 1–7. https://doi.org/10.1016/j.cag.2017.07.027.
[53] M. MacKlin, M. Müller, N. Chentanez, T.Y. Kim, Unified particle physics for real-time applications, ACM Trans. Graph. 33 (2014) 1–12. https://doi.org/10.1145/2601097.2601152.
[54] M. Mu, B. Heidelberger, M. Hennix, J. Ratcliff, Position based dynamics, J. Vis. Commun. Image Represent. 18 (2007) 109–118. https://doi.org/10.1016/j.jvcir.2007.01.005.
[55] M. Müller, B. Heidelberger, M. Teschner, M. Gross, Meshless deformations based on shape matching, ACM Trans. Graph. 24 (2005) 471–478. https://doi.org/10.1145/1073204.1073216.
[56] B. Lovrovic, Z. Mihajlovic, Rigid Body Joints in Real-Time Unified Particle Physics, in: Eurographics 2018, 2018. https://doi.org/10.2312/egs.20181031.
[57] A. Salvadori, G. Del Frate, M. Pagliai, G. Mancini, V. Barone, Immersive virtual reality in computational chemistry: Applications to the analysis of QM and MM data, Int. J. Quantum Chem. 116 (2016) 1731–1746. https://doi.org/10.1002/qua.25207.
[58] M. Wiebrands, C.J. Malajczuk, A.J. Woods, A.L. Rohl, R.L. Mancera, Molecular Dynamics Visualization (MDV): Stereoscopic 3D Display of Biomolecular Structure and Interactions Using the Unity Game Engine, J. Integr. Bioinform. 15 (2018) 1–8. https://doi.org/10.1515/jib-2018-0010.
[59] S.J. Bennie, K.E. Ranaghan, H. Deeks, H.E. Goldsmith, M.B. O'Connor, A.J. Mulholland, D.R. Glowacki, Teaching Enzyme Catalysis Using Interactive Molecular Dynamics in Virtual Reality, J. Chem. Educ. 96 (2019) 2488–2496. https://doi.org/10.1021/acs.jchemed.9b00181.
[60] I. Leap Motion, Controller — Leap Motion JavaScript SDK v2.3 documentation, (n.d.). https://developer-archive.leapmotion.com/documentation/javascript/api/Leap.Controller.html.
[61] J. Löwe, H. Li, K.H. Downing, E. Nogales, Refined structure of αβ-tubulin at 3.5 Å resolution, J. Mol. Biol. 313 (2001) 1045–1057. https://doi.org/10.1006/jmbi.2001.5077.
[62] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, T.E. Ferrin, UCSF Chimera - A visualization system for exploratory research and analysis, J. Comput. Chem. 25 (2004) 1605–1612. https://doi.org/10.1002/jcc.20084.
[63] Blender, (n.d.). www.blender.org (accessed November 4, 2020).
[64] K. Salisbury, F. Conti, F. Barbagli, Haptic rendering: Introductory concepts, IEEE Comput. Graph. Appl. 24 (2004) 24–32. https://doi.org/10.1109/MCG.2004.1274058.
[65] P.A. Kelly, Solid Mechanics Part I: An Introduction to Solid Mechanics, in: Solid Mech. Part 1, 2015: pp. 283–342. http://homepages.engineering.auckland.ac.nz/~pkel015/SolidMechanicsBooks/Part_I/BookSM_Part_I/10_Viscoelasticity/10_Viscoelasticity_Complete.pdf.