Invited Review

A Virtual Reality Computational Platform Dedicated for the Emergence of Global Dynamics in a Massive Swarm of Objects

Gregory GUTMANN*, Ryuzo AZUMA*, and Akihiko KONAGAYA**

Emergence is one of the key design principles in molecular robotics which aims to construct an artifact by means of the bottom-up fabrication of biomolecules as living things. In order to promote rational design of emergence necessary for molecular robotics, we have developed a virtual reality （VR） computational platform. The computational platform incorporates VR particle simulation, VR live control and interactive VR user interface running on GPU computing. The VR particle simulation is capable to emerge global dynamics in a massive swarm of objects represented by a compound of particles interacted by Lennard-Jones potential. The VR live control enables us to tune simulation parameters while running VR particle simulation. The interactive VR user interface enables us to intervene in simulation with virtual hands. GPU computing plays an essential role in achieving more than 90 frames per second in rendering while computing particle simulations in the millions.
Keywords : Emergence, VR particle simulation, VR live control, Interactive VR user interface, Molecular robotics

1. Introduction

Emergence is one of the key design principles in molecular robotics.[1,2] The goal of molecular robotics is to construct an artifact with senses and intelligence by means of the bottom-up fabrication of biomolecules as living things. In case of bottom-up fabrication, emergence of higher order system-level functionality from a self-organized swarm of component objects with limited functionality is essential. In order to study the rational design of emergence necessary for molecular robotics, we have developed a virtual reality （VR） computational platform to emerge global dynamics in a massive swarm of objects.

Particle simulations have been intensively studied to occur emergence in a swarm of objects on computer so far. One of such examples is the "bird-oid object" referred as boids[3] and Vicsek flocking algorithm.[4] Boid follows some simple rules : separation, alignment and cohesion in order to steer to avoid crowding local flockmates, to steer towards the average heading of local flockmates, and to steer to move toward the average position of local flockmates. In Vicsek flocking algorithm, each individual follows simple equations to adjust its speed and turning angle to the neighbor of the individual within a ranging parameter.

Although both boids and Vicsek flocking algorithm are useful for mimicking the behavior of a swarm of objects, our early experiments on microtubule gliding simulation[5,6] indicated the importance of structural design of individual objects as well as the design of interaction forces among the objects for the reproduction of global dynamics. Therefore, we represent an individual object as a compound of particles connected by springs.

For example, a microtubule, a tubular polymers of tubulin that form cytoskeleton in a cell, can be represented by a chain of particles. A microtubule sometime overrides another microtubule when they come across in microtubule gliding assay. Our microtubule gliding simulation system can emerge "overriding of microtubules" from the detailed, smoothly curved circular motion caused by the convergence effects of particle-particle interactions across the chains. Furthermore, the explicit interactions within and between the particles of the chains enable us to maintain realistic microtubule movement including flows and rings throughout long simulation runs. These additions make the particle simulation more complicated but they enable us to produce heterogeneity in object-object interactions, which turns out to be essential in the emergence of global dynamics as seen in experiments.

Our early experience also indicated the limitation of a conventional 2D display to observe global dynamics emerged from the swarm of objects, especially when the number of objects becomes large. This led our research to the VR simulation, an integration of VR and simulation. VR offers a

* School of Computing, Tokyo Institute of Technology.
4259, Nagatsuta, Modori, Yokohama 226-8502, Japan

new way of human-computer interaction. At first glance, a VR system seems to be a set of goggles with small screens. But once mounted display for virtual reality, such as VIVE made by HTC Corporation, is put on, and entering the VR home space, it no longer feels like in the same place as prior to putting on the headset. We believe this sensation of being transported to another space is beneficial to understanding simulations. It creates an immersive experience that allows us to move and look around naturally, removing the need to think about using normal PC controls. The view field becomes the virtual environment, leading to a greater awareness of what is being simulated. Also, due to the immersion and ease of use, it is likely to promote a much greater degree of interest which is very beneficial for learning and understanding, the goal of simulations.

While VR provides many benefits such as the ones listed above, it also comes with strict performance requirements. This is due to possible motion sickness at lower frame rates. Currently, 90 frames per second (FPS) is the accepted minimum frame rate, about 11 milliseconds. It should be noted that 11 millisecond is a very short time window for both the computation work and the visualization of the results from the viewpoint of conventional high performance computing on general purpose graphics processing units (HPC GPGPU). Aside from 3D graphics, most numerical simulation work is batch-based, submitting work and returning later when it is finished. However, the batch-based HPC GPGPU techniques cannot be applied to VR simulation directly, because the smaller the job which is submitted, the harder it becomes to hide the overhead when using the GPU card for computational work.

With regards to scientific simulations, there are typically two main categories of simulations, fine-grained models which strive to represent everything explicitly to attempt to reach the greatest possible level of accuracy, and coarse-grained models which incorporate hypothesis or assumptions to observe the behavior of models. For the former, in case of conventional molecular computation such as molecular dynamics (MD) and molecular orbital (MO), simulation runs can take anywhere from hours to days. Thus, these computational models are not suited for VR aside from visualization after simulation runs. Coarse-grained models, due to the simplifications and smaller scales, with respect to the number of objects, typically have a higher update frequency. This higher update frequency is preferable to realize smooth visualization which would make this a good candidate for VR. However, the lower detail often results in lower accuracy, and the smaller scales that lead to the higher update frequency are still smaller than our target simulation scale.

Attempting to meet both types of performance demands listed above is a sizable challenge, being forced to maintain a high rate of simulation for VR, and requiring large-scale simulations which contain large computational workloads. To meet this goal, the focus of our work has primarily been on how to scale the simulation system across hardware while mitigating the overheads involved with GPU, and removing the limitations of rendering and networks.

The primary objectives for the VR simulation are : (1) a method of computing tens of millions of interacting objects within an 11 millisecond time window, (2) a system that can render hundreds of thousands of objects in VR, and (3) a way to smoothly display visuals when the simulation size reaches into the tens of millions of objects.

Our computational platform also features VR live control, interactive VR user interface and GPU computing. VR live control enables us to change simulation parameters while computing simulations. This facility plays an essential role in searching an optimal parameter set which emerges global dynamics. An interactive VR user interface enables us to intervene into the simulation process. This facility works well when dealing with bio molecules. GPU computing plays an essential role in both VR rendering and simulation performance. The VR particle simulation requires more than 90 frames per second in VR rendering while simulating millions of particles. Various GPU programming techniques are used to make use multiple GPUs as well as a large number of GPU cores.

The organization of this paper is the following. Firstly, section 2 describes technologies adopted in the VR simulation with regards to particle simulation, live control, interactive interface and GPU computing, respectively. Next, section 3 demonstrates some applications including a microtubule gliding assay and DNA origami. Lastly, section 4 concludes the current status and future works of VR simulation.

2. Technologies

The VR computation platform incorporates VR particle simulation, VR live control, interactive VR user interface and GPU computing in the following.

2.1 Representing VR objects in particles

The VR particle simulation consists of two layers : A swarm of compound objects and a network of particles to represent each compound object. In a compound object, particles are connected by springs to form the structure of the compound object. Each particle has its own interaction force defined by Lenard-Jones potential. VR simulation is performed by updating all particle locations according to the forces available from the connected particles with springs and from the neighbor particles with Lenard-Jones potential.

Fig. 1 depicts a typical particle representation of microtubule objects. The microtubule objects are created by connecting a series of particles into a one-dimensional chain using
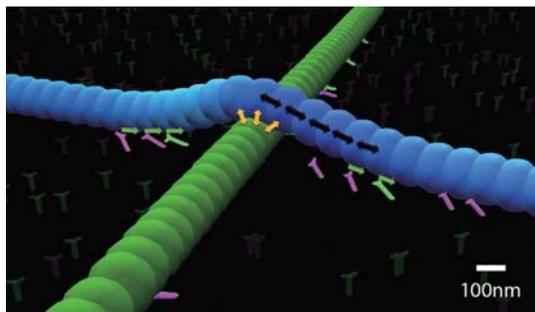
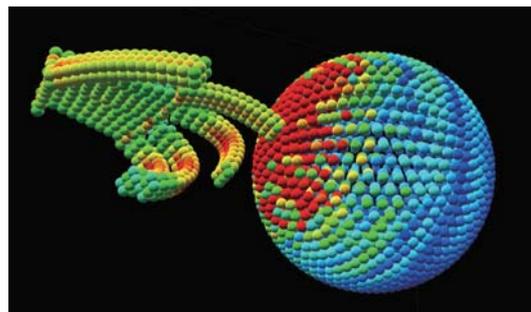**Fig. 1** Representation of microtubules and their particle-particle interaction.

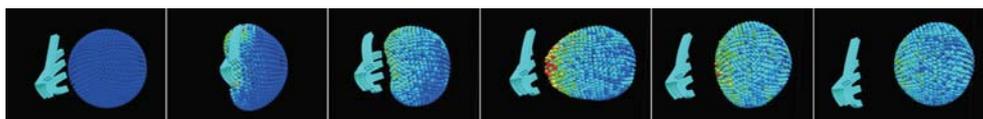**Fig. 2** Visualizing forces when touching a particle sphere.

Fig. 3 Reformation of sphere object when hit by a VR hand.

various forces, which forms a passive object. Then we use particle-like objects to represent motor proteins, which are active objects that seek out microtubule chains and traverse them ; applying forces to the microtubule in the process, which results in microtubule motion. When there are tens of thousands of microtubules which are put into motion by this process, they begin to interact with each other and produce global dynamics like what can be seen in wet experiments.

Using the Lennard-Jones Potential for the particles in the system gives each particle physical properties such as : a physical size from repulsive force, collisions, and a weak attractive force. With respect to microtubules, the repulsive force from Lennard-Jones potential during collisions between particle chains, with specific dynamical properties, is the primary factor which leads to the emergent behavior seen in wet experiments. The attractive component of the Lennard-Jones potential leads to more tightly coupled microtubule motion over time in comparison to only using repulsive forces. This leads to global dynamics of collective motion and emergent behavior.

2.2 Controlling simulation parameters while running

Live control enables us to change simulation parameters while running a simulation. The live control plays an important role in finding interesting global dynamics in a massive swarm of objects. This is because the live control enables us to find "control parameters" which govern the behavior of VR particle simulation while running. It is not too much to say that live control would be the only way to find an optimal parameter set when meta-search algorithms such as genetic algorithm cannot be applied due to the lack of an evaluation function. It should be also noted that most global dynamics are temporal. This might be one of reasons why it is difficult to define an evaluation function for emergence.

Live control is also useful to check robustness with regards to parameter sensitivity. It often happens that some parameter set shows interesting global dynamics in very short ranges of parameters. We do not adopt such a parameter set that does not reflect real world phenomena or experiments.

In the current VR particle simulation, following live control parameters are available :
-Interaction cutoff range
-Lennard-Jones potential : strength, equilibrium distance, scaling of attractive force, etc.
-Bounded and unbounded simulation space
-Energy dissipation（drag/dampening for objects）
-Gravitational force : enable/disable and strength
-Spring mass and strength
-VR hand : adjustable size（particle counts）
-Object color coating based on selected force

VR interface for live control is one of interesting research issues remained as future works. There are many ways to show a control panel in VR. Gesture motion is one of sophisticated user interfaces applicable to VR particle simulation, although it becomes complicated when considering VR interactive interface with virtual hands.

2.3 Touching and sensing VR objects

Interactive VR user interface has a great advantage to apply VR simulation to real world applications if intuitive touching and sensing devices such as virtual hands are provided. The usefulness of interactive VR user interface is demonstrated in **Fig. 2** and **3**. Our virtual hands are represented by a collection of particles reflecting the motion of user's hands detected by Leap motion controller, a commercially available image sensor with two cameras mountable to a VR head mounted display. Each particle has its color representing force level received from neighbor particles. Warm color shows higher force level.

GUTMANN・AZUMA・KONAGAYA : A Virtual Reality Computational Platform
Dedicated for the Emergence of Global Dynamics in a Massive Swarm of Objects

650

Such color change visualizes some kind of interaction forces between VR objects to users（see Fig. 2）. Deformation of VR object is also performed by calculating the interaction power from VR hands to VR object（see Fig. 3）.

Our work has many similarities with interactive molecular dynamics（IMD）, both having interaction and visualization for biological simulations. There have been a number of works in the past that focus on creating IMD software systems.[7–11] In this research field, IMD has been found to be highly beneficial due to the ability to guide reactions to understand how the desired result can be reached, greatly reducing the simulation trials needed amongst other benefits.[7] IMD has also been used as an educational tool to experiment and gain intuition on molecular dynamics.[8] One of the earlier works in IMD created a 3-tier system, which later works took similar approaches, using a computer for haptics, a computer for visualization, and a computer or parallel computational system for simulation.[9] At the time the work was published, they had tested simulating on a dual-core CPU（Central Processing Unit）, in a later work by Dreher, et.al. then scaled up IMD simulations to be run with large compute clusters.[10] In the work by Dreher, et. al., they were able to use a large number of CPU nodes to simulate into the millions of atoms, but due to the distributed nature, this approach is not suitable for VR, because update rates and latency do not meet the minimum requirements for VR. To help reduce the overhead incurred when distributing work across large numbers of nodes, we have taken an approach similar to the work by Luehr. et.al., using a set of massively parallel GPUs all on a single node.[11] This results in having the parallel power much more closely tied together, many parallel processors on a single card and many cards connected by PCI-E. Despite this advantage due to the nature of their simulation, it still ran at rates insufficient for VR. Thus a new approach was needed for our VR simulation.

Both entertainment and professional applications, such as games or computer-aided design（CAD）software, have opted to use game engines as a way to integrate their software into VR. Games typically use a small number, hundreds to thousands, of detailed objects within each scene and a number of advanced effects. This works for most CAD scenarios as well because 3D designs often have similar demands. Since largescale simulations do not contain similar rendering workloads to games, if using a game engine, rendering performance of the scene is never likely to meet the performance needed for VR, primarily due to differences in optimization. Several visualization tools for molecular dynamics simulation results such as VMD,[12] however, at the time of developing our system, there were none that met both our performance requirements and HTC VIVE support. More importantly, developing our own rendering engine is crucial because of our approach to use performance gains to reach our
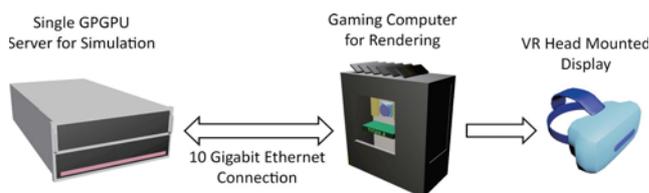

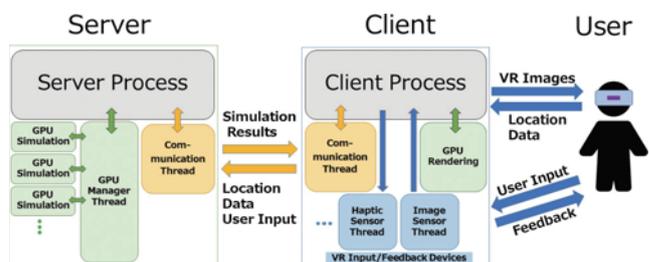**Fig. 4** Client and server type VR computational platform hardware.

Fig. 5 VR simulation workflow on server and client computers.

goal of simulating larger biological systems ; we need to ensure the whole system can be optimized, fully utilizing the hardware and leaving no unneeded operations.

2.4 Enabling large-scale computational simulation to run VR rates

In contrast to CPU which had only incremental performance gains over the years, GPU performance have been increasing at impressive rates which has enabled VR simulation. High-end gaming GPU are capable of 14 Tera FLOPS（FLoating-point Operations Per Second）of FP32（32 bits single precision Floating Point number）, which is an immense amount of performance in comparison to CPU. In case of server and client type VR computational platform, we adopt gaming computer with 1-2 GPUs for rendering and GPGPU servers with 10 GPUs for simulation（**Fig. 4**）. Since, each GPU（NVIDIA Titan X）can incorporate 3584 CUDA Cores, highly parallel thread processing techniques is needed to obtain high performance in VR rendering and live simulation. We also intensively use asynchronous communication between CPU and GPU, GPU and CPU, and client node and server node, to increase VR simulation performance.

Fig. 5 shows the VR simulation workflow running on the client and server computers. The client process is in charge of VR rendering and interactive VR user interface with an image sensor and a controller. The server process is in charge of particle simulation with GPUs. Both client and server processes communicate with each other through the communication threads. The client process receives location data and user input from the VR head set and controller, then sends them to the server process. The server process receives the input data, computes the simulation step, and sends back simulation

results to the client process. Then, the client process produces VR images from the simulation results.

Since modern GPUs consist of thousands of cores, the GPU cores are clocked lower than CPU cores but due to their number, GPU offer much more performance. At the time of writing, many of the high-end GPUs now advertise over 10 Tera FLOPS of performance. GPU cards also contain several types of tightly coupled memory including registers, caches and RAM. The key to GPGPU computing is efficient memory management needed for computation both locally on the GPU and between the GPU and CPU memory. There are many low-level optimization technique for individual GPU kernels which is vital for efficiently using the GPU, however this paper focuses on higher level algorithms which aim at efficiently using many GPUs concurrently.

For multi-GPU computation, data movement is the primary performance bottleneck such as transferring results off of the GPU or GPU-GPU communication. The next issue is scheduling what task should be done on each GPU. If the task sizes being computed on each GPU can be kept constant that is a big help ; however, that is often not possible. Thus, scheduling the tasks to be done by each GPU becomes a second performance bottleneck. Most current commercial applications, such as photo or video editing, use an offloading technique, where the application does some initial processing on the CPU including setting up the GPU task, then sends the task to the GPU and waits, and then continues the post processing on CPU and finishes or repeats the cycle. Recent multi-GPU research papers cover many algorithms for various applications which can fully utilize multiple GPU to solve problems.[13, 14] However, while they efficiently use GPU(s), the problems being solved typically are very large and take hours to days. To our knowledge multi-GPU systems have not yet been applied for the use in interactive VR simulations.

Our GPU computing solutions for the interactive VR simulation are the following.

（1） Reducing CPU side workload for task management : CPU becomes the limiting factor even though it is only doing a fraction of the total simulation work if the CPU is in charge of task management for GPU. Moving a large portion of the task management from CPU to GPU enables us to use more GPU cards without performance degradation.

（2） Reducing GPU-GPU communication overhead for data management : To help negate the overhead associated with data copying to the GPU and off the GPU, we use data persistence. Each GPU in our system is initialized with the full data set of GPU sided data, in contrast to distributed computing with distributed memory. Note that this technique does put a limitation on data scaling since increasing the GPU count increases parallel performance with space partitioning method but not total data size.

With our current model when using 10 GPU cards, the maximum simulation size is about 6-12 million interactive particles depending on simulation models. This is large enough to represent a VR scene surrounding a user. Without using data persistence at this scale, the simulation update rate is not sufficient for VR and not very desirable to sit and watch on a screen, even if we use conventional high performance programming technique such as double buffering and overlapping of data movement and computation, to name a few. Advanced GPU-GPU network such NVLink is helpful but limited in scalability at the time of writing.

By using data persistence, we can dramatically reduce the data traffic not only between GPU and GPU but also GPU and CPU. As for GPU-GPU data transfers, the only data moved among GPUs is the location of particles which are crossing the boundary of space partitions. Data transfer can be minimized if all GPU have the same object properties in their GPU memory in advance. The data needed for visualization which is usually just a subset of particles. Therefore, conventional GPU-GPU connections such as a PCI-E bus is scale enough up to 10 GPUs when data persistence technology is used.

3.　Results and discussion

Our VR simulation platform can emerge various kinds of global dynamics. We simply show its effectiveness by demonstrating microtubule gliding pattern formation and DNA origami simulation in this paper.

3.1　Microtubule gliding pattern formation

Microtubule gliding assay is a biological experiment where molecular motors are fixed to a glass surface with microtubules above in a liquid solution. When adenosine triphosphate （ATP） is added to this setup, the molecular motors attach to the microtubules and walk across for a uni-direction propelling the microtubule across the surface. Swarm behavior such as flows and rings can be observed in the microtubule gliding assay when the appropriate molecular interaction condition is provided.[15]

Within our simulation, we have taken an approach which allows for a variable level coarse-grained model when forming microtubules out of particle chains. When aiming to simulate with higher detail, we use each particle as a 25 nanometer diameter particle reflecting the average diameter of a microtubule in wet experiments. As a result, microtubules that are 5, 10 and 15 micrometers would consist of 200, 400, 600 particles connected by bonds or springs, respectively. The strength of the spring force is estimated so that these particle chains can match with the elastic properties of microtubules in wet experiments.

The motion of microtubule objects strongly depends on its length, speed and elasticity represented by spring force between particles. Interestingly, more than two thousand
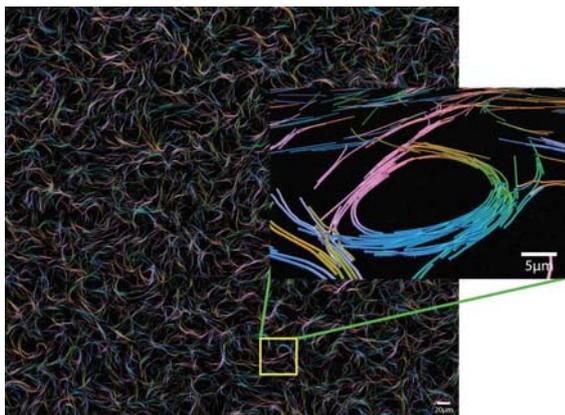
Fig. 6 Observed microtubule gliding motion patterns on particle simulation.

Fig. 7 Represented DNA origami structures with more than 450 thousand atoms in VR simulation.

microtubule chains are required for emergent global dynamics to begin to appear. We believe the number is strongly related to the simulation space scale and the lengths of the microtubule particle chains. The greater number of microtubules, the greater probability to form interesting features such as flows and rings. In **Fig. 6**, an image taken from a simulation run with 50,000 microtubules can be seen. This suggests that complexity of particle-particle interaction may contribute to the emergence of global dynamics.

3.2 VR DNA origami structure visualization after molecular dynamics simulation

DNA origami is one of DNA nano technologies used in molecular robotics. DNA origami self-organizes a 2D or 3D structure in nano scale by means of the DNA hybridization of a scaffold DNA strand and hundreds of staple DNA strands.[16] An M13 bacteriophage circular DNA strand with 7249 bases is often used as a scaffold DNA strand. Short DNA strands typified with 32 bases are used for staple DNA strands. The shape of DNA origami is strongly dependent on the design of the staple DNA strand sequences. DNA hybridization, that is, self-organization of DNA strands plays an important role in DNA origami construction.

Prediction of atomic-level DNA origami structure on a computer is one of the hot topics in the field of molecular robotics. All atom DNA origami MD simulation[17] can predict the atomic-level molecular structures in details. We also use interactive VR simulation for the visualization of DNA origami structures simulated by the all atom MD simulation. **Fig. 7** shows the snap shot of the DNA origami, named "angry mark" which contains more than 450 thousand particles.

In case of all atom DNA origami simulation, DNA origami is too large to deal with the interactive VR simulation because more than three weeks are required to observe long range dynamics in MD simulation when using a super computer. In ord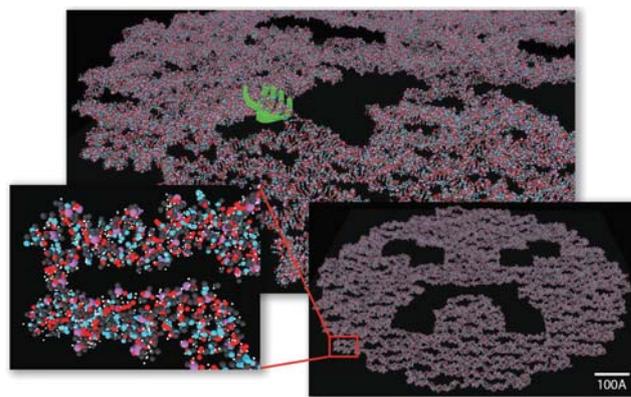er to deal with this issue, we adopted a hybrid approach in which conventional MD simulation for long time dynamics and interactive VR simulation for the visualization of the result of MD simulation.

Nevertheless, the interactive VR simulation plays an important role in the analysis of the MD simulation result. This is partly because of the increase of viewing angle to observe the specific DNA double strand formation while viewing whole DNA origami image, partly because of interactive touching interface with virtual hands. It should be noted that the impression of DNA origami look very differently when it is moving in the VR simulation.

4. Conclusions

This paper reviews the current status of the VR particle simulation research dedicated for the rational design of molecular robotics with regards to the emergence of global dynamics in a massive swarm of objects. The VR particle simulation is promising but still in its early stage. It would be a practical tool applicable to real world applications if an appropriate user interface is provided. As a future works, there are many possible avenues to improve the user interface ; a haptic feedback devise is one of such activities that is strongly demanded especially when touching biomolecules.

5. Acknowledgement

GUTMANN・AZUMA・KONAGAYA：A Virtual Reality Computational Platform
Dedicated for the Emergence of Global Dynamics in a Massive Swarm of Objects

653

References

1) S. Murata, A. Konagaya, S. Kobayashi, H. Saito, and M. Hagiya, "Molecular Robotics : A New Paradigm for Artifacts," New Generation Computing, Ohmsha, Ltd. and Springer, **31**, pp. 27-45, (2013).

2) M. Hagiya, A. Konagaya, S. Kobayashi, H. Saito, and S. Murata, "Molecular Robots with Sensors and Intelligence," Accounts of Chemical Research, **47**, pp. 1681-1690 (2014).

3) T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel Type of Phase Transition in a System of Self-Driven Paticles," Physical Review Letters, **75**, pp. 1226-1229, (1995).

4) C.W. Reynolds, "Flocks, Herds, and Schools : A Distributed Behavioral Model," Computer Graphics, **21** (SIGGRAPH '87 Conference Proceedings), pp. 25-34, (1987).

5) G. Gutmann, D. Inoue, A. Kakugo, and A. Konagaya, "Real-Time 3D Microtubule Gliding Simulation, Life System Modeling and Simulation," Communications in Compuer and Information Science, ICSEE 2014, LSMS **461**, pp. 13-22, (2014).

6) G. Gutmann D. Inoue, A. Kakugo, and A. Konagaya, "Parallel Interaction Detection Algorithms for a Particle-based Live Controlled Real-time Microtubule Gliding Simulation System Accelerated by GPGPU," New Generation Computing, **35**, pp. 157-180, (2017).

7) P. Grayson, E. Tajkhorshid, and K.D. Schulten, "S Mechanisms of Selectivity in Channels and Enzymes Studied with Interactive Molecular Dynamics," Biophysical Journal, **85**, pp. 36-48, (2003).

8) D.V. Schroeder, "Interactive molecular dynamics," American Journal of Physics, **83**, pp. 210-218, (2015).

9) J.E. Stone, J. Gullingsrud, and K. Schulten, "A system for interactive molecular dynamics simulation," proceedings of the 2001 symposium on Interactive 3D graphics-I3D'01, pp. 191-194, (2001).

10) M. Dreher, M. Piuzzi, A. Turki, M. Chavent, M. Baaden, N. Ferey, S. Limet, B. Raffin, and S. Robert, "Interactive Molecular Dynamics : Scaling up to Large Systems," Procedia Computer Science, **18**, pp. 20-29, (2013).

11) N. Luehr, A.G. Jin, and T.J. Martinez, "Ab Initio Interactive Molecular Dynamics on Graphical Processing Units (GPUs)," Journal of Chemical Theory and Computation, **11**, pp. 4536-4544, (2015).

12) W. Humphrey, A, Dalke, and K. Schulten, "VMD : Visual molecular dynamics," Journal of Molecular Graphics, **14**, pp. 33-38, (1996).

13) J. Domínguez, A. Crespo, D. Valdez-Balderas, B. Rogers, and M. Gómez-Gesteira, "New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters," Computer Physics Communications, **184**, pp. 1848-1860, (2013).

14) J. Zhou, Y. Cui, E. Poyraz, D.J. Choi, and C.C. Guest, "Multi-GPU Implementation of a 3D Finite Difference Time Domain Earthquake Code on Heterogeneous Supercomputers," Procedia Computer Science, **18**, pp. 1255-1264, (2013).

15) D. Inoue, T. Nitta, A.M.R. Kabir, K. Sada, J.P. Gong, A. Konagaya, and A. Kakugo, "Sensing surface mechanical deformation using active probes driven by motor proteins," Nature Communications, 7 : 12557, pp. 1-10, (2016).

16) P.W.K. Rothemund, "Folding DNA to create nanoscale shapes and patterns," Nature, **440**, pp. 297-302, (2006).

17) R. Azuma, S. Kishi, G. Gutmann, and A. Konagaya, "All-atom molecular dynamics of film supported flat-shaped DNA origami in water," Chem-Bio Informatics Journal, **18**, pp. 96-118, (2018).

Greg GUTMANN

He received his B.S from John Carroll University, Ohio, USA, in 2013. He received both his M. Eng. and PhD from TITECH in 2015 and 2016. Then, he started his work as an Assistant Professor in TITECH. His research interest includes GPU programming, interactive simulation, Deep Learning, Virtual reality, and anything that may seem like an interesting challenge. He has worked on Molecular Robotics and Artificial Muscle projects within the Konagaya Laboratory in TITECH, and as of 2018 he is an Nvidia University Ambassador.

Azuma Ryuzo

He received his B.S from Keio University, Yokohama, Japan, in 1997. He received both his M. Sci. and PhD from University of Tokyo in 1999 and 2001. Then, he worked as a postdoc researcher in RIKEN in 2001, Kyushu Institute of Technology in 2008, Osaka University in 2012, and TITECH in 2017. He has worked on computational physics, bioinformatics, electron microscopy, and molecular robotics projects. His research interest includes computational science and methodologies for micro and nanoscale simulations. As of October 2018 he has become a specially appointed assistant professor in TITECH.

Akihiko KONAGAYA

He received his M.Sci. and D.Eng. from Tokyo Institute of Technology (TITECH) in 1980 and 1995, respectively. Then, he entered ; NEC Corporation in 1980, JAIST in 1997, RIKEN GSC in 2003 and TITECH in 2009, respectively. He is a professor of Tokyo Institute of Technology since 2009. After graduation, he has researched and developed for the interdisciplinary field of informatics, biology and chemistry including bioinformatics and molecular robotics, to name a few.