



Original software publication

Game-Engine-Assisted Research platform for Scientific computing (GEARS) in Virtual Reality



Brandon K. Horton^a, Rajiv K. Kalia^{a,b,c,d}, Erick Moen^{b,e}, Aiichiro Nakano^{a,b,c,d,f}, Ken-ichi Nomura^{a,d,*}, Michael Qian^a, Priya Vashishta^{a,b,c,d}, Anders Hafreager^g

^a Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0241, USA

^b Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0241, USA

^c Department of Physics & Astronomy, University of Southern California, Los Angeles, CA 90089-0241, USA

^d Department of Materials Science and Chemical Engineering, University of Southern California, Los Angeles, CA 90089-0241, USA

^e Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-0241, USA

^f Department of Biological Science, University of Southern California, Los Angeles, CA 90089-0241, USA

^g Department of Physics, University of Oslo, 0316 Oslo, Norway

ARTICLE INFO

Article history:

Received 26 April 2018

Received in revised form 21 November 2018

Accepted 11 January 2019

Keywords:

Virtual reality

Real-time simulation visualization

Molecular dynamics

LAMMPS

ABSTRACT

The Game-Engine-Assisted Research platform for Scientific computing (GEARS) is a visualization framework developed at the Materials Genome Innovation for Computational Software (MAGICS) center to perform simulations and on-the-fly data exploration in virtual reality (VR) environments. This hardware-agnostic framework accommodates multiple programming languages and game engines in addition to supporting integration with a widely-used materials simulation engine called LAMMPS. GEARS also features a novel data exploration tool called virtual confocal microscopy, which endows scientific visualization with enhanced functionality.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

Current code version	v1.0
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_42
Legal Code License	GPL 3.0
Code versioning system used	git
Software code languages, tools, and services used	C++, C#, Python
Compilation requirements, operating environments & dependencies	Windows 10, Visual Studio, Unity, Unreal Engine, Python, LAMMPS
If available Link to developer documentation/manual	https://github.com/USCCACS/GEARS/blob/master/README.md https://magics.usc.edu/gears/
Support email for questions	cacs@usc.edu

1. Motivation and significance

Data visualization plays a key role in scientific discovery. Though quantitative analysis is indispensable, researchers are often forced to apply summary metrics blindly. Unfortunately, some of these statistics are limited in their ability to describe the system under test and can be misleading, as illustrated by Anscombe's Quartet [2].

* Corresponding author at: Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0241, USA.
E-mail address: knomura@usc.edu (K.-i. Nomura).

Visualization software such as VisIt [3], ParaView [4], Visual Molecular Dynamics (VMD) [5], and OVITO [6] provide a straightforward interface to create three-dimensional (3D) images and observe patterns in the output of simulations. Though often used among experienced researchers, it remains challenging to extract information from datasets that consist of complex geometries or contain a large number of entities because of the intrinsic limitation of two-dimensional (2D) representations of 3D objects for traditional displays. Immersive 3D visualization technology like ImmersaDesk, Tile Wall, and CAVE2 [7] provide unique spaces for collaboration and scientific communications [8]. These solutions, however, require specialized knowledge of the respective systems

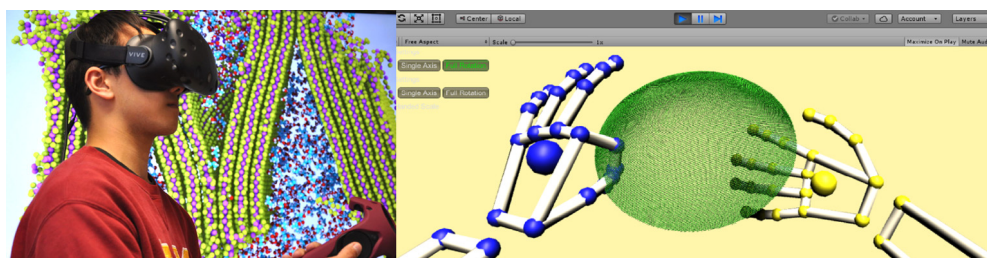


Fig. 1. (Left) A user wearing a head-mounted display. (Right) A user manipulating simulation data of a tellurine material [1]. The skeletal hands are processed by the LeapMotion library in the Unity 5 Game Engine. Using finger pinches and gesture controls, the user can rotate, translate, and scale the structure intuitively in immersive VR. The surrounding graphical user interface is the Unity editor.

and clear expectations of the simulation's outcome. Furthermore, the high costs of these systems make them prohibitive for most researchers.

An affordable alternative is an easily-accessible, agnostic software platform designed for the increasingly available head-mounted displays (HMDs) developed for consumer virtual reality (VR). VR is a powerful visualization tool that has grown steadily in popularity over the past two decades, fueled by the success of HMDs like Oculus Rift [9] and HTC Vive [10]. Despite its widespread adoption for video games and media applications, as well as its potential as the most immersive and intuitive method for viewing data, VR has been underutilized by the scientific community. To transform VR into a common scientific tool, we have developed a software framework, called Game-Engine-Assisted Research platform for Scientific computing (GEARS) that facilitates the adoption of VR technologies and allows researchers to take advantage of the unique analytical advantages that the medium offers (Fig. 1).

2. Software description

2.1. Interactive data visualization

GEARS makes use of commodity game engines, like Unity [13] and Unreal Engine [14], to simplify access to VR headsets. The most straightforward application of VR to data visualization is interactive viewing of pre-computed results for data exploration. To realize this feature, GEARS employs the iBET workflow [15], in which external 3D modelers, such as VMD and Blender [16], are used to create a 3D object that can be added to the scene along with the appropriate script (for example, LeapRTS.cs for the Leap Motion controller [17]) (see Fig. 2).

This aspect of GEARS allows users a quick, straightforward outlet for immediate visualization of snapshots of data from materials simulations or molecular structures. Though the current interactive visualization mode supports single frame data, we plan to expand this feature by creating multiple scenes containing different simulation time steps for more dynamical data exploration. Fig. 3 showcases several examples of GEARS interactive visualizations used in computational researches.

2.2. Real-time simulation visualization

GEARS also takes advantage of the programming capabilities provided by game engines, such as C# and JavaScript supported by Unity and C++ for Unreal Engine, to facilitate reuse of existing simulation codes. This mode of GEARS is suitable to explore simulation results in real-time, rendered entirely within the game engine.

To realize this real-time rendering, it is critically important to design an efficient data-bridging method between the game engine and user-developed simulation program. GEARS employs two approaches, called *Run-when-Ready* and *Render-when-Ready*, depending on the size of the data and the complexity of the simulation engine. *Run-when-Ready* calls the simulation engine upon a

frame update to advance the state of the simulation (for example particle positions) of the frame by one timestep. *Render-when-Ready* makes use of the multi-threading optimization technique to offload the simulation engine computation onto a new thread while the main thread is only responsible for handling the render state of the game engine. In this approach, when the simulation engine finishes one timestep computation, the main thread either updates the frame state or stores the simulation states for rendering later while the worker thread continues to produce new states as background. Using this technique as well as other optimizations, like impostors and GPU instancing [18], we have demonstrated real-time simulation sizes reaching up to 500,000 particles. To minimize the amount of coding necessary for GEARS users, we provide two demos – Lennard-Jones molecular dynamics (MD) [19] and electron transfer simulation by kinetic Monte Carlo [20] – implementing the *Run-when-Ready* and *Render-when-Ready* approaches in the GEARS GitHub repository. See Demo3 of UnityGEARS on the GEARS repository.

In an effort to make our immersive scientific computing suite accessible to a broad research community, we have also interfaced GEARS with one of the most widely used MD simulation engine called LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [21]. LAMMPS was developed at Sandia National Laboratory and supports a variety of interatomic potentials, statistical ensembles, and flexible simulation setups. The LAMMPS interface enables users to visualize their MD simulations in VR environment without any coding. A “How-to” for setting up the LAMMPS integration environment and an example demo are documented in detail in the Lammmps Compilation section of UnrealGEARS on the GEARS repository.

2.3. Virtual confocal microscopy

To fully realize the promise of immersive scientific computing, we have developed a new tool to enhance GEARS' visualization capability, called virtual confocal microscopy. Confocal microscopy has become an indispensable biomedical tool, and it has proven useful in many other scientific fields [22–25]. It allows for increased resolution, contrast, and optical sectioning while minimizing photodamage and bleaching concerns. The technique uses point illumination and a pinhole to scan the sample and reject out-of-focus light. As a result, the image only contains information very close to the focal plane, offering greater resolution [24]. An analogous approach in VR could enable more methodical investigations of simulated complex structures.

Virtual confocal microscopy utilizes Unity's surface shader capabilities to control how each vertex on the structure is rendered in order to highlight certain areas or planes of the material in the simulation. Our solution seeks to generate a viewing plane that sits in front of the user's head, follows their head movement, and always maintains a set distance from the user. This distance, as well as the thickness of the highlighted viewing plane can be specified

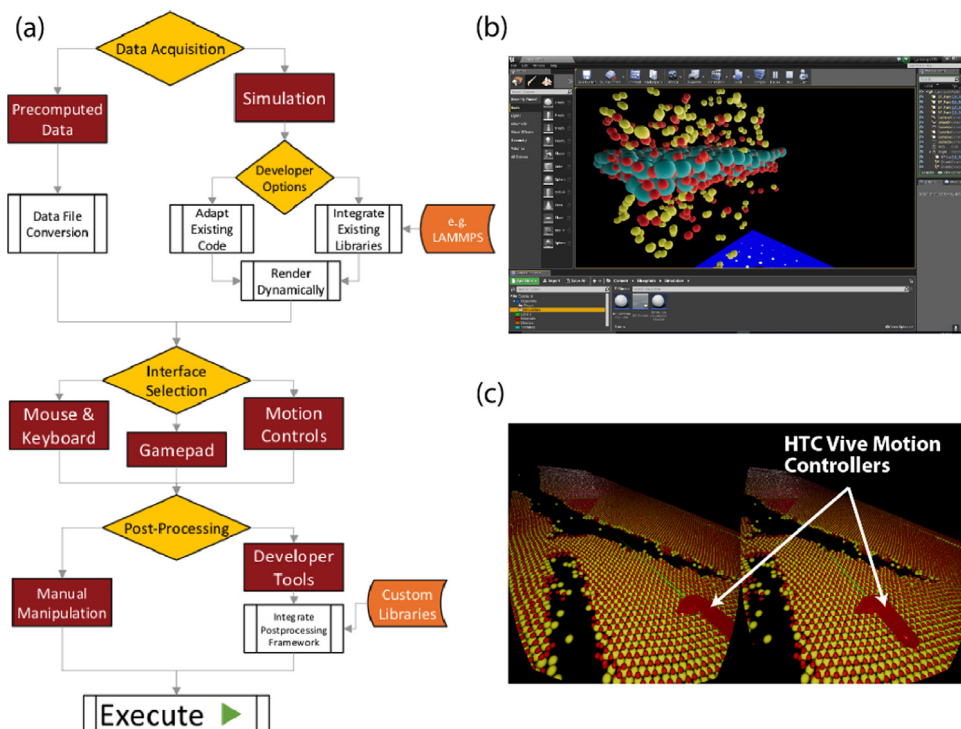


Fig. 2. (a) Three primary steps (Data Acquisition, Interface Selection, and Post-Processing) of the GEARs visualization workflow. In the Data Acquisition step, a researcher supplies data to the engine by either converting pre-computed results to a particular format (e.g. .obj, .fbx) or computing data on the fly via simulation. In the Interface Selection step, a GEARs user selects the method or interface through which they desire to interact with their pre-computed data or real-time simulation. In the Post-Processing step, a researcher selects a post-processing technique, which can either utilize the control scheme they chose in the Interface Selection step or make use of custom code. (b) An example of visualizing a chemical vapor deposition (CVD) simulation [11,12] through GEARs' LAMMPS integration. (c) Simulating and exploring data related to a molybdenum disulfide (MoS₂) fracture via the LAMMPS integration. Using the controllers indicated by white lines, researchers can freely navigate through the structure, pause and resume the animation, and even highlight regions of interest with the provided laser attachment (shown in green).

and changed dynamically by the user. The rest of the simulated system will stay mostly transparent (with the opacity also dictated by user input), except for the vertices that intersect with this plane. In this way, the user can scan through complicated geometries using just their head movements - a unique experimental advantage only possible in VR.

2.4. Control schemes

A carefully designed control scheme is essential to facilitate interaction between the user and various data representations within the immersive simulation. These interactions range from object manipulation and spatial translations to temporal scaling. It is critical that these interactions mimic natural motion as much as possible to limit user discomfort, especially as it relates to user motion within the environment [26–28]. GEARs supports input from a variety of sources that includes the Leap Motion controller, Oculus Touch, and HTC Vive motion for immersive environments, as well as keyboard and mouse for small and precise adjustments to aspects of the simulation.

3. Illustrative examples

Recently, researchers performed MD simulations of desalination membranes to understand and characterize how local, atomic structures contribute to macroscopic behavior [31]. The relevant dimensions of the polyamide membranes of interest range from a tenth of a nanometer to 100 nm, making it an excellent candidate for computational studies to extract optimization information not easily accessible experimentally. They have found that water molecules permeated the membrane through benzene rings and

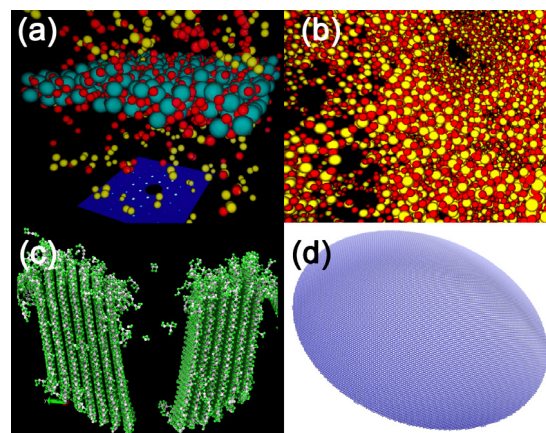


Fig. 3. Examples of interactive visualization: (a) Realtime rendering and simulation of MoS₂ synthesis. (b) Walkthrough of three-dimensional nanometric cage structure in a porous amorphous silica. (c) Exfoliation process due to dynamic shear flow on a 2D material surface [29]. (d) Strain-induced structural transformation in an atomic telluride layer [30].

that the degree to which these monomers were cross-linked governed the speed of permeation along various paths [31]. However, the construction of these membranes is very complex, making a quick identification of preferential paths difficult with traditional 2D visualizations. The ability to section such a complicated super-structure in VR environment with simple head motion helps researchers identify prevalent substructures of interest in these simulated materials, and will greatly accelerate the research production cycle (see Fig. 4).

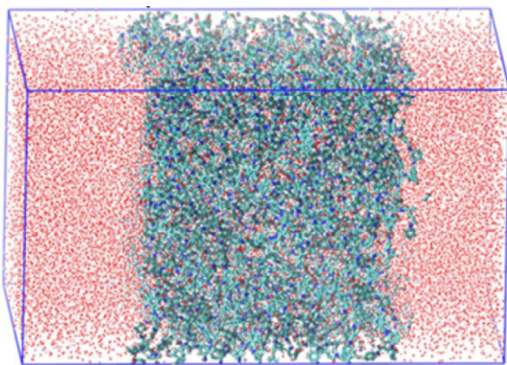


Fig. 4. Crosslinked aromatic polyamide membrane immersed in water. Water molecules (red) diffuse through nanometric channels in the membrane (green) at the middle of the simulation box.

Source: Adapted with permission from [27].

© 2016 American Chemical Society.

4. Impact

Physical simulations, like MD, inherently exist in 3D space, yet modern, desktop viewing platforms tend to only support 2D visualizations. GEARS, as an open-source, commodity platform, makes exploratory visualization and immersive data analysis more accessible for researchers. Using our workflow, outlined above, and our provided vertex shader, one can apply 3D virtual confocal microscopy to any discrete data set. Alternatively, should a researcher need to create their own customized shader (depending on their investigative needs), they can make use of the sample environments we have provided to productively bring their own data to an interactive, immersive environment. Consequently, this manuscript sets a foundational codebase for scientists to make use of modern graphics shaders for experimental data analysis.

Even if shader technology or virtual confocal microscopy is not a primary goal, researchers can still make use of the immersive playback capabilities in GEARS. For pre-computed simulations, this generally means pausing animations then exploring and analyzing sequestered regions. However, not only have we provided an immersive visualization engine for researchers, but also a simulation platform that encapsulates much of the complicated logic necessary for real-time data analysis. Researchers can choose to either port their existing data (e.g. LAMMPS DUMP files) or run entirely new simulations on-the-fly. With our framework, LAMMPS scripts can be written with VR visualization accounted for by design. Researchers can even pause a simulation mid-run, giving them opportunities to make adjustments to their simulation before supplying new commands to the linked LAMMPS library through our code interface. Afterwards, they can continue running the simulation and note the effects of their adjustments to the system. This option for immersive simulation steering is just one example of how our software also acts as a platform for VR-oriented extensions in LAMMPS. By integrating LAMMPS with Unreal, much like Atomify [32] integrated the MD simulator with mobile operating systems, we have improved the framework's extensibility as a research tool. Additionally, build upon our sample environments, researchers can more productively take advantage of the engine's rich features — for instance, its native server framework for multi-user LAMMPS VR collaboration, built-in blueprint system for interactive UI, and plugin controls for immersive navigation of a 3D simulation.

GEARS in Unity and Unreal act as an extensible hybrid simulation and visualization engine. It has already seen use in classroom and research settings, acting as a new medium for presenting data. This medium allows researchers to take advantage of humanity's

innate spatial awareness to improve its user's productivity as well as a new avenue for scientific communication. It has facilitated the exchange of simulation data to both experts and non-experts, most notably at U.S. Department of Energy (DOE) Materials Genome Innovation for Computational Software (MAGICS) workshops and at a DOE conference in Washington DC.

5. Conclusions

We have developed a hardware-agnostic visualization framework called GEARS that takes advantage of the unique possibilities and power associated with the use of VR environments and rapidly evolving game engine technology to explore scientific datasets. VR has become a powerful new method of engaging with users in a more tactile, visceral way. The resurgence in hardware to support VR has resulted in a rich software environment that allows developers access to these tools with minimal investment. While VR technologies have a myriad of applications, they could be particularly useful for scientific data visualization and exploration, collaboration with domain experts, as well as communications with non-scientific audiences. With GEARS, we provide various approaches and modalities for researchers to interactively explore their simulation data, from visualization of pre-computed datasets in VR to the integration of external simulation engines such as LAMMPS. GEARS even allows for researchers to take advantage of immersive data visualization and simulation without requiring additional coding. Driven by the multimedia and gaming industries, it is likely that VR platforms will continue to enhance user functionality and interactivity at lower costs, making them even more approachable to researchers. Therefore, the novel data exploration and collaboration capabilities GEARS offers will be broadly beneficial for scientific community.

Acknowledgments

This work was supported as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award Number DE-SC00014607, for our MAGICS (Materials Genome Innovation for Computational Software) center, <https://magics.usc.edu>.

References

- [1] Apte A, et al. Structural phase transformation in strained monolayer MoWSe₂ alloy. *ACS Nano* 2018;12(4):3468–76.
- [2] Anscombe FJ. Graphs in statistical analysis. *Amer Statist* 1973;27(1):17–21.
- [3] Childs H, et al. VisIt: An end-user tool for visualizing and analyzing very large data. 2012.
- [4] Ahrens J, Geveci B, Law C, Hansen CD, Johnson CR. An end-user tool for large-data visualization. In: *The Visualization Handbook*. 2005, p. 717.
- [5] Humphrey W, Dalke A, Schulten K. VMD: Visual molecular dynamics. *J Mol Graph* 1996;14(1):33–8.
- [6] Stukowski A. Visualization and analysis of atomistic simulation data with OVITO—the open visualization tool. *Modelling Simulation Mater Sci Eng* 2010;18(1).
- [7] Febretti A, et al. CAVE2: A hybrid reality environment for immersive simulation and information analysis. In: *Engineering Reality of Virtual Reality 2013*, vol. 8649. 2013.
- [8] Manjrekar S, Sandilya S, Bhosale D, Kanchi S, Pitkar A, Gondhalekar M. CAVE: An Emerging Immersive Technology – A Review. *IEEE*; 2014, p. 131–6.
- [9] Oculus Rift. <https://www.oculus.com/rift/>.
- [10] HTC Vive. <https://www.vive.com/us/>.
- [11] Hong S, et al. Computational synthesis of MoS₂ layers by reactive molecular dynamics simulations: initial sulfidation of MoO₃ surfaces. *Nano Lett* 2017;17(8):4866–72.
- [12] Hong S, et al. Chemical vapor deposition synthesis of MoS₂ layers from the direct sulfidation of MoO₃ surfaces using reactive molecular dynamics simulations. *J Phys Chem C* 2018.
- [13] Unity. <https://unity3d.com>.
- [14] Unreal Engine 4 Documentation. <https://docs.unrealengine.com/latest/INT/index.html>.

- [15] Nakano CM, et al. iBET: Immersive visualization of biological electron-transfer dynamics. *J Mol Graph Modelling* 2016;65:94–9.
- [16] Blender. <https://www.blender.org>.
- [17] Leap Motion Developers. <https://developer.leapmotion.com>.
- [18] Tarini M, Cignoni P, Montani C. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Trans Vis Comput Graphics* 2006;12(5):1237–44, (in English).
- [19] Frenkel D, Smit B. *Understanding Molecular Simulation*. second ed.. San Diego, CA: Academic Press; 2001.
- [20] Byun HS, El-Naggar MY, Nakano A, Vashishta P. A derivation and scalable implementation of the synchronous parallel kinetic Monte Carlo method for simulating long-time dynamics. *Comput Phys Comm* 2017;219:246–54.
- [21] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. *J Comput Phys* 1995;117(1):1–19.
- [22] Franck C, Hong S, Maskarinec SA, Tirrell DA, Ravichandran G. Three-dimensional full-field measurements of large deformations in soft materials using confocal microscopy and digital volume correlation. *Exp Mech* 2007;47(3):427–38.
- [23] Rajadhyaksha M, Grossman M, Esterowitz D, Webb RH, Rox Anderson R. In vivo confocal scanning laser microscopy of human skin: Melanin provides strong contrast. *J Invest Dermatol* 1995;104(6):946–52.
- [24] Webb RH. Confocal optical microscopy. *Rep Progr Phys* 1996;59(3):427–71.
- [25] Diaspro A. *Confocal and Two-Photon Microscopy: Foundations, Applications and Advances*. Wiley; 2011.
- [26] Slater M, Usoh M, Steed A. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Trans Comput-Human Interact* 1995;2(3):201–19.
- [27] Templeman JN, Denbrook PS, Sibert LE. Virtual locomotion: walking in place through virtual environments. *Presence: Teleoperators Virtual Environ* 1999;8(6):598–617, (in en).
- [28] Wann JP, Rushton S, Mon-Williams M. Natural problems for stereoscopic depth perception in virtual environments. *Vis Res* 1995;35(19):2731–6.
- [29] Zhou G, et al. Molecular simulation of MoS₂ exfoliation. *Sci Rep* 2018;8(1):16761.
- [30] Amey A, et al. Polytypism in ultrathin tellurium. *2D Materials* 2019;6(1):015013.
- [31] Wei T, et al. Aromatic polyamide reverse-osmosis membrane: An atomistic molecular dynamics simulation. *J Phys Chem B* 2016;120(39):10311–8.
- [32] Atomify - a real time LAMMPS visualizer. <https://github.com/ovilab/atomify>.