Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

CyberEarth: a Virtual Simulation Platform for Robotics and Cyber-Physical Systems

Xiaoyang Zhang and Hongpeng Wang, Jingtai Liu.
Tianjin Key Laboratory of Intelligent Robotics
College of Artificial Intelligence
Nankai University
Tianjin, 300350, China

Haifeng Li
Department of Computer Science and Technology
Civil Aviation University of China
Tianjin, 300300, China

Abstract— The increasing sophisticated robot and intelligent system applications require universal visualization platforms which can guarantee the security and efficiency of task process execution in the situation of user-programming and using different kinds of automated equipment. In this paper, we present a universal visualization framework to build up program-driven simulation software of complex robots and intelligent systems by integrating several open-source technical modules, including Ubuntu Linux operation-system, QT Creator IDE environment, ROS robot operation system, OSG(OpenSceneGraph) 3D scene, osgEarth GIS(Geographic Information System)-based 3D scene, and also Python based user-programing robotic script language. Many complex visualization simulation systems of complex tasks in wide area and dynamic scenarios are realized by using this framework. Based on this framework, we built a virtual simulation platform CyberEarth for robotics and Cyber-Physical systems. The typical robotic simulation task, which is a visual coverage task for Multi-Agent/UAV, is also introduced to demonstration the universality of this platform.

Index Terms— Virtual Simulation; Open Source Architecture; 3D Visualization; Intelligent System; User Programmable

I. Introduction

The robotic simulation plays a fundamental role of testing the models and techniques in a controlled environment prior to conducting experiments on real physical agents. Open Source, user-programmable, and universal are key cores for researchers to build their own simulation environments. However, Building simulation environments can be a tiresome and complex task. In this paper, we propose an open source user-programmable framework that can enable end users to build their own robotic simulation platforms. The middle ware of the system consists of the robot operating system (ROS). ROS [1] is an open source operating system designed for robots. It provides features such as hardware abstraction, low-level device control, packages for common functionalities, a communication framework, and a variety of libraries and tools.

Virtual simulation technology is an integrated innovation technology supported by a variety of high and new technologies including artificial intelligence technology, which aims to create a real-time three-dimensional virtual world that reflects the change and interaction of entities and objects for users in the way of simulation.

It can describe the “human, machine and environment” virtually, break through the constraints of reality, time, space and funds, and construct a complex and dynamic intelligent simulation system. It can also be regarded as the “Visualization of Artificial Intelligence”.

Virtual reality and intelligent simulation are important components of virtual simulation. They are related to each other and support each other.

El-Shawa et al. [2] combines physical reality with virtual reality, then presents a comparison framework that may be extended to other HRI studies investigating the relationship of different parameters between PR and VR. Pecka et al. [3] has presents a novel technique that allows for both computationally fast and sufficiently plausible simulation of vehicles with non-deformable tracks and has been implemented as a plugin for the open-source physics-based simulator Gazebo using the Open Dynamics Engine. This paper [4] focuses on human interaction and introduces a simple prototype of a lightweight sensing and tactile communication system that allows two-way communication between two people or between a person and a computer. There is a direct object manipulation system that a user can manipulate objects constrained to each other by his or her hands in paper [5].

In addition, [6]-[10] focus on simulation and [11]-[15] follow virtual reality with interest. In general, they all combine the virtual world with the real world through virtual simulation and solve specific problems.

* This work is supported by National Natural Science Foundation of China (Grant No.61973173, 91848108), Technology Research and Development Program of Tianjin (Grant No.18ZXZNGX00340)
(Corresponding author: Hongpeng Wang, hpwang@nankai.edu.cn)

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

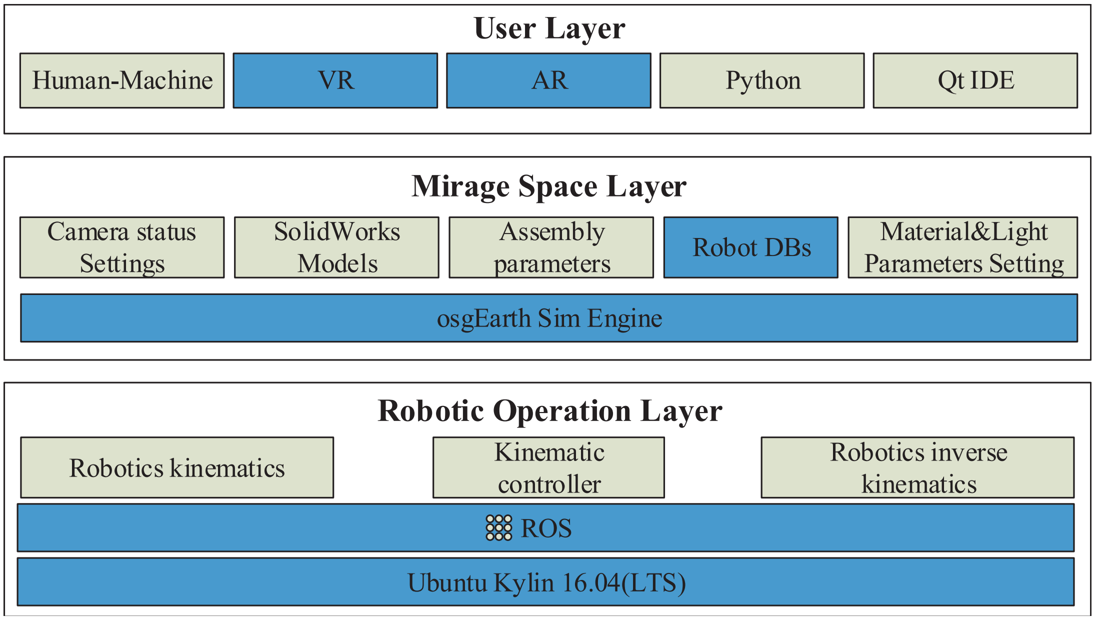

Fig. 1. Base Framework for Robotics and Cyber-Physical Systems. The corresponding parts of these blue blocks mainly depend on integrating open source systems and SDKs, and the corresponding parts of these gray blocks are mainly developed by ourselves.

At the moment, many complex tasks of intelligent systems, such as field monitoring, rescue and disaster relief, military dispatch in wide-area complex scenarios, lack of effective simulation research tools.

Commercial virtual simulation system development software in the market has its own emphasis, but it is difficult to achieve full support for GIS simulation, AI algorithm, AR/VR human-computer interaction, remote control, large data visualization.

It's important for us to refine the common key technologies in virtual simulation, such as intelligent modeling, driver and human-computer interaction, and build an open source, open and universal virtual simulation engine and development platform based on standardization and modularization.

II. Architecture

In order to offer simulation and supervision to the robotics, intelligent systems and cyber-physical systems, we research it based on the following technical routes.

A. OpenSource Framework

As shown in Fig. 1, the framework interacts with real robots through ROS at the robotic operation layer; the mirage space layer is developed based on osgEarth, an open source GIS simulation engine, to realize the co-simulation of geographic information system and robotic system; the upper user layer can interact with users through VR and AR devices, and be manipulated by scripts written by users.

Among these layers, the mirage space layer can combine the 3D-GIS virtual earth model with the local area simulation scene. Users can intuitively experience 3D simulations from macro to micro. In the macroscopic earth model rendering, it supports the use of massive data loading such as satellite cloud maps, elevation maps, and vector maps, and manages GIS data using xml format files to achieve dynamic loading. In the local area simulation, it supports the import of 3D models built by mainstream modeling software such as Solidworks and 3DMAX. It can be controlled by simple scripts, and can also be manipulated by real data through ROS.

B. Configurable Environment

In order to build an open source virtual simulation platform, the compiler environment of OSG and osgEarth is first configured. Then the Qt Creator is used for development, which can easily realize cross-platform publishing. The compiler environment is configured in the virtual machine for easy portability and backup.

Compared with industrial standard OpenGL or other graphics libraries, the basic rendering engine OSG used in this platform has great advantages. Apart from being open source and cross-platform, the engine encapsulates and provides a large number of algorithms to improve the runtime performance of programs, direct data interfaces for almost all mainstream data formats, including paging databases, and support for scripting language systems

Python and Tcl, which will break through the ultimate limitation of human-computer interaction performance of existing interactive systems.

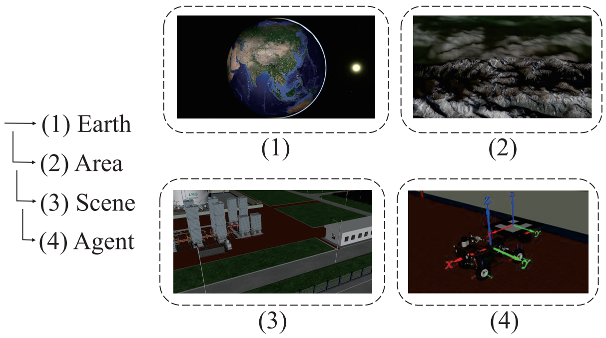

Fig. 2. Grading Display of Virtual Data in CyberEarth. Users can smoothly zoom between the four levels.

C. Technical Principle

The platform provides a surface that can express vast space on which satellite images and elevation data can be superimposed. The earth's surface is composed of triangles subdivided by wireless, and the algorithm provides a method to dynamically manage the density of triangles.

When advancing from space to the ground, the grids on the earth's surface will become denser and denser, and the terrain expressed will become more and more delicate. When advancing to the ground, the maximum accuracy will be achieved. When observing in space, the fluctuation of the earth's surface will be neglected, and the grids on the earth will be greatly simplified.

Similar changes have also been made to the images appended to the earth's surface, which are constantly switching between images of different resolutions.

A specific local area on the earth can be put into a virtual scene. In order to fully reproduce the running state of the robot in virtual scene, it is necessary to receive the whole state of the robot (such as the position and posture of the planar mobile robot [x, y, θ], the joint angles of the manipulator, the position and posture of the space mobile robot [x, y, z, α, γ, ϕ], etc.). We can see the progressive relationships of these levels in Fig. 2.

After the virtual simulation platform obtains the whole state of the robot, it can calculate the pose matrix of each module by using the TF transform tree inherent in the robot model, so that each module can be placed in a suitable pose in real time to realize the kinematics simulation of the robot.

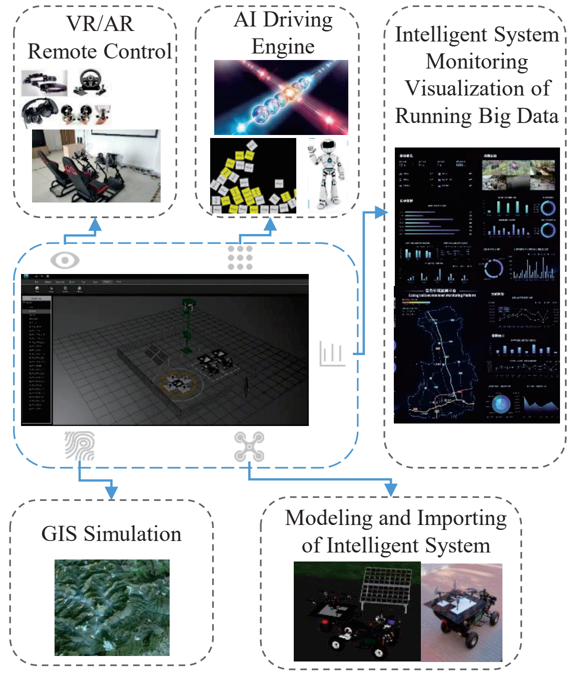

Fig. 3. Functional Display of Platform Characteristics. The platform can support these parts by integrating them together.

D. Platform Characteristics

Based on the open source framework, we independently developed a user programmable real-time virtual simulation engine and development platform. As shown in Fig. 3, the platform supports GIS simulation, AI algorithm, AR/VR human-computer interaction, remote control and large data visualization.

It provides a vast space of the earth surface, on which satellite images and elevation data can be superimposed, and mainstream vector data such as KML and SHP can also be superimposed. At the same time, three-dimensional scenes can be loaded. Three-dimensional scene is edited in the editor, which can load multiple three-dimensional sub-scenes to form massive three-dimensional data.

It can be tightly integrated with physical robots through ROS. In wide-area geographic information scenarios, virtual robots can be manipulated by users through AR/VR devices, and real robots can be manipulated remotely by AI algorithm. VR and AR are currently undergoing secondary development for tasks.

Readers can download the platform of CyberEarth from https://vr.nankai.edu.cn/CyberEarth/ and perform secondary development by attaching scripts written for specific tasks with different programming languages, such as Python, Simulink/MATLAB, etc.

Compared to other frameworks, such as Gazebo and Unity, CyberEarth achieves better simulation results on specific tasks, such as the visual coverage task for multi-agent/UAV. CyberEarth can effectively use a large amount of GIS data for simulation of this task.

E. Hardware Requirements

The CPU should be on i7 or above, the graphics card uses NVIDIA GTX 1070 or better that memory is larger than 4G, the running memory is larger than 8G, the hard disk capacity is greater than 1T, and the screen resolution is not less than 1920 1080.

We tested it on the HP Z240 graphics workstation with a frame rate of up to 30FPS, which reflected that the real-time performance was guaranteed.

III. Platform Modules

The simulation platform Cyberearth is mainly composed of the following modules and other task-specific extension modules.

A. Scene Design Module

The module is responsible for integrating all satellite cloud images, elevation images, and local scenes to be loaded, and storing the final design results in xml format for subsequent use.

The platform provides a three-dimensional earth with a basic world layer. Users can superimpose more detailed satellite imagery and elevation images of a local area according to their own needs.

In the partial scene design, the user can select the file import from the platform model library, or import the model of Solidworks or 3DMAX design from the outside. Then, by right-clicking on the model, the pose is adjusted with a simple button operation, and the direct input fixed angle, Euler angle, quaternion, and fourth-order pose transformation matrix are supported to accurately adjust the pose. The designed partial scene as a whole can be placed in a certain area by setting the latitude and longitude, or by setting the pose matrix in the planetary coordinate system.

B. Collaborative Rendering Module

The module is controlled by the calculation engine based on OpenCV development by default, and the user can control the movement of the model through simple script editing.

At the same time, the platform provides a real-time file stream interface, allowing users to design their own motion algorithms using Python or Simulink/MATLAB which uses the real-time file stream to send pose information to the platform. The platform strictly follows this information for rendering.

C. Real-time Measurement Module

In this module, the model can be selected by right-clicking, and then the basic information can be measured and displayed in the background in real time, such as longitude, latitude, altitude, pose in the planetary coordinate system, and pose under the local coordinate system.

After that, users can choose different export methods to save the information. Some different types of format are supported in the module. For example, binary data, text files, xml files and yaml files.

IV. Example of Application

We take the visual coverage task for multi-agent/UAV as an application example. Through virtual simulation and supervision on our platform, we can optimize the system design, improve the system technical scheme and screen the task operation.

A. Task Definition

Area Coverage Path Planning (CPP) of UAV is defined as that how to avoid obstacles and hazards, then plan an optimal route for vision sensor detection area to traverse the area to be covered under the premise of satisfying some optimal performance indicators.

Multi-UAV cooperative coverage path planning means that multiple UAVs traverse the coverage area by coordinating with each other, so that the overall performance index can be optimized.

Because of its parallelism, the multi-UAV system can improve the efficiency of area coverage and shorten the time to complete the task. When a single UAV fails, the remaining UAV can still complete the task. Therefore, it will have better ability of task execution and fault tolerance.

B. Task Significance

In urban, field and other wide-area, dynamic, unstructured environments, there is an increasing demand for efficient and timely patrol and monitoring.

With the increasing demand for eco-environmental protection, the patrol of the wild natural environment has more and more monitoring tasks for various animals and plants. Such patrol tasks are tedious and time-consuming, time-consuming and labor-intensive.

In a wide-area, dynamic, and unstructured environment, airborne robots can load intelligent camera-based detection equipment from an aerial perspective with high-precision, high-speed, self-planned motion, and exert a large viewing angle and vertical displacement. Such advantages have gained wide application in the field of environmental monitoring.

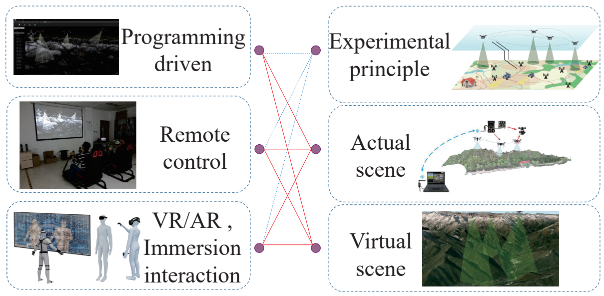

Fig. 4. Principal Display of Platform Application. The user can implement the right part in the platform after completing three parts on the left side.

C. Applied Principle

The base principle of application is shown in Fig. 4.

Through using USB communication interface, the remote control system is connected with the virtual simulation platform. A set of driving control rules are formed by modeling the kinematics and dynamics of the virtual robot system.

According to the changes of each state in remote control system, the state of virtual robotic system is updated in real time to remote control and immersion experience. Then intelligent algorithm is designed to complete the task of visual coverage path planning.

D. Application Method

Before the aerial robot performs the coverage task, it is necessary to test the feasibility of the task. CyberEarth can realize the scene display of wide area, complex scene and intelligent system. Users can design tasks through scripts.

In order to meet the specific coverage requirements, it is necessary to plan the path points of aerial robots in advance. However, we haven't know whether the planned path meets the kinematics and dynamics requirements of UAV. As a result, it is also necessary to test and adjust through the virtual simulation platform. Finally, we will get the executable scheme by CyberEarth.

When aerial robots perform coverage tasks, the virtual simulation platform can monitor and track the motion status of aerial robots in real time to ensure that the robot can safely and reliably perform the given tasks.

In case of dangerous situation, such as low battery and malfunctioning vision sensor, users can get relevant information through the simulation platform at the first time, and make timely response measures.

While monitoring and tracking robots, the virtual simulation platform will record the simulation data and upload the information to the cloud for subsequent data analysis and scene reproduction. The cloud data can be used to reproduce the robot and the covered area, so that the analysis and comparison of the task completion can be made, and the satisfaction index of the task completion can be obtained.

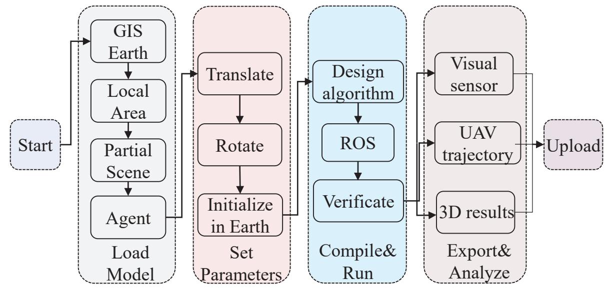

Fig. 5. Process of CyberEarth Application in Virtual Simulation. After completing the four large boxes, users can upload the data to the designated server for verification by the server's computing power.

E. Operation Steps

As shown in Fig. 5, We can do simulation by following steps in CyberEarth.

① Start the simulation platform, build new projects, load digital earth and large-scale geographic information images and elevation data.

② Import the environment static body model, control each module to carry out translation, rotation and other operations to build the UAV mission scene.

③ Import or select the multi-agent/UAV moving body model under the system model tree, and customize the parameters.

④ Assign related values through the graphical operation interface to define the initial position and attitude states of the experimental scenes and the agents/UAVs.

⑤ Design and implement UAV visual coverage path planning algorithm by graphical/script programming combined with ROS robot operating system.

⑥ Compile and run the program to verify the effectiveness and safety of UAV mission execution by assessing the risk of the task in CyberEarth.

⑦ Record and export the program-driven visual sensor information, UAV trajectory, scene monitoring and reconstruction results.

⑧ Package and upload the compiled program to the designated server to verify the principle and technology of the actual UAV experiment.

In fact, it is possible to perform a complete simulation verification of the dynamics, kinematics, and safety of the UAV trajectory locally. However, it takes a lot of time. Therefore, the user can use the server's computing power by uploading the data to the specified server.

F. Experimental Result

The Fig. 6 shows three four-rotor UAV performing visual coverage tasks on the virtual simulation platform. The red point cloud, the green point cloud, and the blue point cloud are the key points of reconstruction detected by three cameras. By performing mathematical operations on these points in the image, the 3D reconstruction results shown in the lower right corner are obtained. From the point cloud density of the results, we find the average density is 25 pixels/point, which means that each point reflects the information of its adjacent 25 pixels.

From the perspective of reconstruction results, the planned trajectory can meet the expected requirements. After that, we will further measure and evaluate the real reconstruction effect in the actual multi-UAV system to complete the coverage task with a better method.

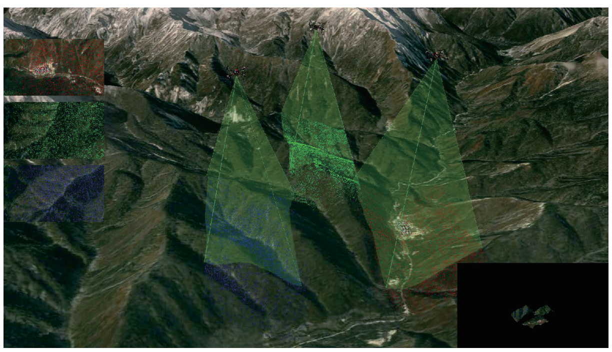

Fig. 6. Experimental Result of 3D Overlay Reconstruction. Three four-rotor UAV perform visual coverage tasks in CyberEarth. The pictures captured by the camera are on the left. The results of three-dimensional reconstruction are shown in the lower right corner.

V. Conclusions

In this paper, we propose a general open source framework, then a user programmable real-time virtual simulation engine and development platform are built based on this framework, which can be used for program-driven simulation, monitoring of complex robots and environments.

In real complex scenarios, it is very demanding to improve the control, planning, perception, cognition, learning, interaction, decision-making and execution capabilities of agents for experimental environment, equipment and materials. With this platform, we can better analyze the motion scheme and control algorithm of the robot, so as to solve the problems existing in the process of robot design, manufacture and operation, avoid accidents and unpredictable losses caused by direct manipulation of the robot entity, find out defects and shortcomings in time, and avoid economic losses.

In the future, we will further improve this framework and platform of CyberEarth to better reflect the "Visualization of Artificial Intelligence".

References

[1] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. "ROS: an open-source robot operatiug system," in ICRA Workshop on Open Source Software, 2009, pp. 1-6.

[2] S. El-Shawa, N. Kraemer, S. Sheikholeslami, R. Mead, and E. A. Croft. " "Is this the real life? Is this just fantasy": Human proxemic preferences for recognizing robot gestures in physical reality and virtual reality," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2017, pp. 341-348.

[3] M. Pecka, K. Zimmermann, and T. Svoboda, "Fast simulation of vehicles with non-deformable tracks," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2017, pp. 6414-6419.

[4] P. A. Kavathekar and D. J. Balkcom, "A tactile shirt for teaching human motion tasks," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2017, pp. 878-885.

[5] J. Kim and J. Park, "Direct hand manipulation of constrained virtual objects," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2017, pp. 357-362.

[6] C. E. Ag ero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. L. Rivero, J. Manzo, E. Krotkov, and G. Pratt, "Inside the Virtual Robotics Challenge: Simulating Real-Time Robotic Disaster Response," IEEE Trans. Autom. Sci. Eng., vol. 12, no. 2, pp. 494-506, 2015.

[7] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., vol. 3, pp. 2149-2154 vol.3, 2004.

[8] P. Gyawali and J. McGough, "Simulation of detecting and climbing a ladder for a humanoid robot," in Proc. IEEE Int. Conf. Electro Information Technology, 2013, pp. 1-6.

[9] W. Yao, W. Dai, J. Xiao, H. Lu, and Z. Zheng, "A simulation system based on ROS and Gazebo for RoboCup Middle Size League," in Proc. IEEE Int. Conf. Robotics and Biomimetics, 2015, pp. 54-59.

[10] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," in Proc. IEEE Int. Conf. Robotics and Automation, 2015, pp. 4397-4404.

[11] W. Usher, P. Klacansky, F. Federer, P. Bremer, A. Knoll, J. Yarch, A. Angelucci, and V. Pascucci, "A Virtual Reality Visualization Tool for Neuron Tracing," IEEE Trans. Vis. Comput. Graphics, vol. 24, no. 1, pp. 994-1003, 2018.

[12] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How Do People Explore Virtual Environments?," IEEE Trans. Vis. Comput. Graphics, vol. 24, no. 4, pp. 1633-1642, 2018.

[13] R. Nagao, K. Matsumoto, T. Narumi, T. Tanikawa, and M. Hirose, "Ascending and Descending in Virtual Reality: Simple and Safe System Using Passive Haptics," IEEE Trans. Vis. Comput. Graphics, vol. 24, no. 4, pp. 1584-1593, 2018.

[14] D. T. Han, M. Suhail, and E. D. Ragan, "Evaluating Remapped Physical Reach for Hand Interactions with Passive Haptics in Virtual Reality," IEEE Trans. Vis. Comput. Graphics, vol. 24, no. 4, pp. 1467-1476, 2018.

[15] S. Ghosh, L. Winston, N. Panchal, P. Kimura-Thollander, J. Hotnog, D. Cheong, G. Reyes, and G. D. Abowd, "NotifiVR: Exploring Interruptions and Notifications in Virtual Reality," IEEE Trans. Vis. Comput. Graphics, vol. 24, no. 4, pp. 1447-1456, 2018.