A Study on Intelligent VR/AR Education Platform for Realistic Content Production

[1]Hyun-Sook Lee, [2*]Jee-Uk Heu

Abstract

In recent years, a platform providing a Visual Programming development environment capable of 3D editing and interaction editing in an In-VR environment to quickly prototype VR/AR contents for education of VR and AR for general users and children. In the past, VR contents were mostly viewed by users. However, thanks to the rapid development of recent computing technologies, VR contents interacting with users have emerged as a device capable of tracking user behavior in a small size It was able to appear. In addition, because VR is extended to AR and MR, it can be used in all three virtual environments and requires efficient user interface(UI). In this paper, we propose UI based on eye tracking. Eye-tracking-based UI not only reduces the amount of time the user directly manipulates the controller, but also dramatically lowers the time spent on simple operations, while reducing the need for a dedicated controller by allowing multiple types of controllers to be used in combination. The proposed platform can easily create a prototype of their intended VR/AR App(or content) even for users(beginners) who do not have a certain level of knowledge and experience in 3D graphics and software coding, and share it with others. Therefore, this paper proposes a method to use VAL more effectively in a 5G environment.

Keywords: *Virtual Reality, Augmented Reality, VR/AR Education, VR/AR User Interface, Intelligence VR/AR App Platform*

I. Introduction

The concept of 3D modeling among computer graphics first appeared in 1976. Although the theoretical basis is similar, the requirements are different depending on the field used, and various commercial tools have been proposed and used. As commercial tools have been advanced and the underlying technology has been developed, the storage contents and mechanisms of each other are different, making them incompatible with each other. Coordinate axes, one of the most basic elements of 3D content(hereinafter referred to as content), have different writing order for each tool, and the method of expressing one object, the position of the initial starting point and reference point of the entire content, and the basic content setting are all different.[1][2][3]

The VR/AR App platform based on Visual Language means a platform that supports the rapid creation of prototypes of VR/AR App using Visual Language in the VR/AR environment[4][5]. The purpose of the platform is to provide a Visual Programming development environment capable of 3D editing and interaction editing in an In-VR environment so that VR and AR content can be quickly created. To this end, we develop a functional language-based “VR/AR-specific language”(hereinafter VAL) and Visual Language, easily implement VR/AR prototypes using these languages on the platform, and then prototype based on the network. You can share and collaborate with others, and you can continue to create content in Unity, Unreal, etc. without interruption with minimal work.

The prototyping platform has the following features so that users (initiators) who have no knowledge and experience of 3D graphics and software coding can easily create and share the prototype of the VR/AR

[1] *Kangnam, University, Dept. of AI.SW Convergence Research Institute, Research Professor (lhsnjhj@kangnam.ac.kr)*
[2*Corresponding Author] *Kangnam University, Dept. of KNU Cham-Injae College, Assistant Professor (jeeukheu@kangnam.ac.kr)*

Received: Dec. 07, 2023, Revised: Jan. 02, 2024, Accepted: Jan. 11, 2024

App(or content) they intend and share with others:

First, the basic composition of 3D objects and scenes in In-VR to provide immersion in 3D VR scenes and object interaction implementation VR/AR is made to be able to be created and modified directly in In-VR by wearing HMD as well as existing 2D screen-based editors. As the purpose of the 3D scene is to construct a 3D scene that can fully express the planning intention of the producer in a short time, the function of helping to quickly complete the desired scene composition through selecting and arranging the already created object(or Prefab) rather than creating a realistic 3D object and making a small correction will be provided.

Second, in describing interaction, which is a key element of VR/AR contents implemented with Visual Language, it provides visual language development environment so that people who have not received coding education can easily achieve their goals. Interaction or effect, which is widely used in the prototyping process, is provided in the form of a pre-made library, allowing the user to proceed prototyping simply by selecting the appropriate interaction/effect and setting the appropriate parameter. In addition, it provides an interaction editor based on a visual language to create an interaction intended by a user, and easily creates a prototype by easily writing and testing a code in a virtual/enhanced reality environment.

Third, the contents developed by the linked prototyping platform as a commercial VR engine can be used in connection with commercial VR engines such as Unity/Unreal. For this purpose, the Visual Language code used for prototyping is converted into a script language of the target VR engine. Prototyping platforms can also produce 3D objects, animations, and interactions, but if you want high-quality games and movie-level effects, you can work more efficiently if you can start making them in prototyping.

Fourth, in order to rapidly progress the VR/AR prototyping based on the function type language the various effect and event etc. are implemented to the simple and efficient function type language as well as the object which user immediately can use. It provides. Functional language is particularly strong in parallel processing, so it is expected to be applied to more complex and various forms of VR/AR prototyping implementation in the future.

Fifth, this platform presupposes sharing with others through network, and for this purpose, it presupposes sharing through 5G network as well as existing network. However, since it requires a considerable amount of data traffic in transmitting 3D objects, it is necessary to consider it and therefore, network connection and control functions need to be implemented.

Users who have no programming experience in this prototype platform will construct a three-dimensional scene that is the background of VR content by wearing VR HMD and arranging bundle-sets(or pre-sets) provided by the platform in a way that seems to match the Lego block in reality. And the interaction of objects in the scene(including players) is implemented by using intuitive visual language in In-VR mode, applying interaction and testing the results immediately. The user can concentrate on the more realistic scene composition and interaction implementation because the interaction creation/correction/execution-test by visual language can be proceeded without interruption with the composition/editing of 3D scene in the In-VR environment.

In this paper, we propose a partition-based discovery scheme for fast device discovery in RDM. In the proposed scheme, all devices in the RDM network are divided into several partitions as per the device UID, and the controller performs the device discovery for each partition by using a suitable response timer for the devices.

This paper is organized as follows: Section 2 discusses the existing device discovery scheme. In Section 3, we propose the fast device discovery scheme for RDM. In Section 4, we analyze and compare the existing and proposed schemes in terms of device discovery times. Section 5 concludes this paper.

II. Related Works

2.1. 5G

5G is a newly proposed mobile communication standard in 3GPP, which has proposed the existing mobile communication standard[6][7][8].

(1) ultra-wideband services (eMBB: enhanced mobile broadband)

(2) High confidence/super low delay communication (URLLC: Ultra Reliable & Low Latency

Communications)
(3) Mass connection (mMTC: Machine-Type Communications)

It is designed to provide a much faster data transmission rate from 100Mbps to a maximum of 20Gbps per user using a wider frequency bandwidth and more antennas to support ultra-wideband services. The beamforming technology, which is applied to this purpose, can control the antenna to concentrate or block radio energy in a specific direction, and can minimize interference between each beam.

High-trust/super low-delay communication is a function that takes into consideration the existing general communication methods such as robot remote control, autonomous driving vehicle sharing the surrounding traffic situation through communication, and real-time interactive game., The goal is to minimize the delay time to 1ms, and to do this, we are optimizing the field of wireless resource management and network design.

In the case of mass connection, it was proposed to cope with the environment in which one user accesses the network with a number of devices or a road full of factory or autonomous driving traffic. Currently, the goal is to support 1 million connections per km.

2.2. VAL(VR/AR Language)

The VR/AR language(VAL) is a technique for describing various VR/AR situations by one standard[9][10][11][12]. Therefore, VR/AR contents described as VAL should be played on the same screen anywhere, and the same interpretation should be done anywhere. Since the complex functions created in the editor can be interpreted differently depending on the analysis engine, it is necessary to change and reconstruct the function of the VR editor to a combination of functions that are simplified as much as possible. It is necessary to study whether these functions have any problems in 5G network environment with importance and connection methods.

2.3. Mobile Edge Computing(MEC)

Mobile Edge Computing (MEC) is one of the latest network technologies, and it is a form of connecting edge computing, an evolutionary type of cloud computing technique, to mobile networks[13][14][15]. Edge computing is a method of processing the processes to be processed by the server in a small server(edge) in a network near the user. This method has the following advantages as the physical distance between the user and the server to handle the user's request is getting closer.

First, the response time is reduced compared to the existing server client method. Of course, the response time is absolutely influenced by the physical distance, so the closer the distance, the less the response time. This is a great advantage for real-time applications that are sensitive to response time, such as VR.

Second, the load is reduced because the process that the server processes is processed in the edge server near the user. Here, the overload refers to both the computing power of the server and the burden on the network traffic. The more users communicate with the server, the more network traffic near the server will increase in proportion to the number of users, and the amount of operation to be processed by the server will also increase. However, if the user requests are processed by the edge server, the user's request is not trans-mitted to the main server, so the burden on the network will be reduced, and the user request processing process that one server has to pay is dispersed. In addition, there are benefits in terms of security and failure response.

2.4. Eye Tracking

Eye tracking technology began in the 1800s with research on human eye movements [9][16], and in 1908 a device was developed to track eye gaze. In the 1980s, a method of manipulating a computer by measuring the user's gaze in real time was suggested. The initial method was a form of eye contact with special contact lenses, which was generally difficult to use [17][18]. Currently commonly used methods calculate the center of the pupil and the reflected light to calculate the direction or point of interest the user is looking at [9][19]. This optical tracking method is the most popular because it is non-invasive and low cost.

2.5. VR/AR Education Platform

On this platform, the user will create VR/AR App prototypes using the VR Editor for beginners and semi-experts. The prototype is searched and played with content created by other users in the platform. In addition, the source code used in the production can be shared among users and collaborated through a web-based shared platform [3][20].

In particular, it supports converting the prototype code created by the user into a format suitable for a commercial VR engine such as unity or unreal, and then linking it with a commercial VR engine to enhance the completeness of the final contents. This helps ensure that the prototype stage and content creation stage do not go away without interruption, improving both the efficiency of the creative work and the completeness of the content.

The system is basically developed on the basis of Microsoft .Net Framework. VAL is selected as the language supported by Microsoft .Net Framework. Windows operating system with .Net framework is supported by various 3D engines for game, and VR/AR development is also supported most actively. Currently, .Net Framework supports other operating systems such as Linux/OS-X through Mono framework, which is an open-source project [21][22][23].

Actor-based AKKA.Net is used as a message infrastructure for efficient management of multiple objects in VR space. Akka.net has already begun to support the .Net framework through Akka.Net as a proven solution in Java/JVM.

The basic structure of the visual programming platform to be implemented on this basis is shown in Figure 1 [24][25]. The 3D Scene Editor is the basic editor of the VR Editor and provides a direct interface to the user.

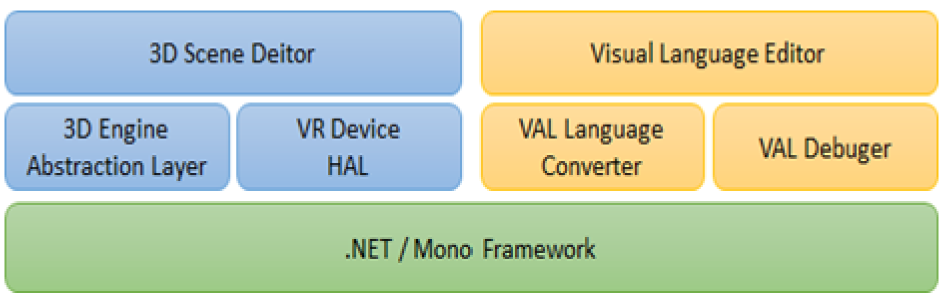

Figure 1. Visual Programming Platform Basic Functionality Configuration

VR Device HAL is a layer abstraction of VR HMD and controller. It supports various VR HMDs and controllers marketed by various manufacturers in the platform. Based on OpenVR and WebVR, the scope of support will be maximized [13][26].

The visual language editor supports dataflow and block coding, and provides an In-VR editor with a 2D editor environment. Considering the usability of visual language editor in VR space with VR HMD, it is most important to implement language expression and editor UI that does not give fatigue and burden to users. Code created/modified by the editor should provide an instant debug mode that can be executed immediately.

Figure 2 shows visual programming platform software configuration. The VR editor is developed based on A-Frame. Visual language is interpreted as VAL through platform server's VR API, converted into JavaScript corresponding to VAL, and sent to VR Editor. This script runs the 3D graphics engine of Three.js/WebGL and provides the VR experience to the user through WebVR Browser. This function configuration is implemented in the following software configuration in the Web-based VR Editor for beginners [27][28][29].

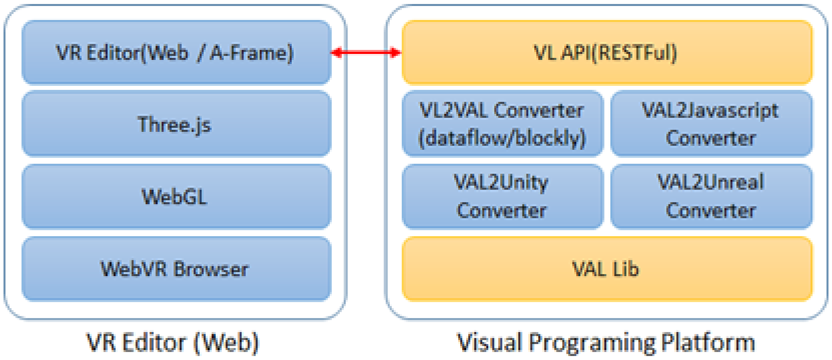

Figure 2. Visual Programming Platform Software Configuration

III. Proposed VR/AR Education Platform

3.1. VR/AR Education Platform

The goal of this platform is to enable users to easily prototype VR/AR App in In-VR environment based on the ability to share code developed in visual language. In order to provide such an In-VR development environment, a VR platform supporting position tracking of the VR HMD is indispensably required.

As shown in Figure 3, it consists of user PC, web server, middleware, and database, which provides to users more easy-to-use VR and AR development environment.

The proposed VR/AR platform enables users (novice users) who do not have some level of knowledge and experience about 3D graphics and software coding to easily create a prototype of their intended VR/AR App (or content).

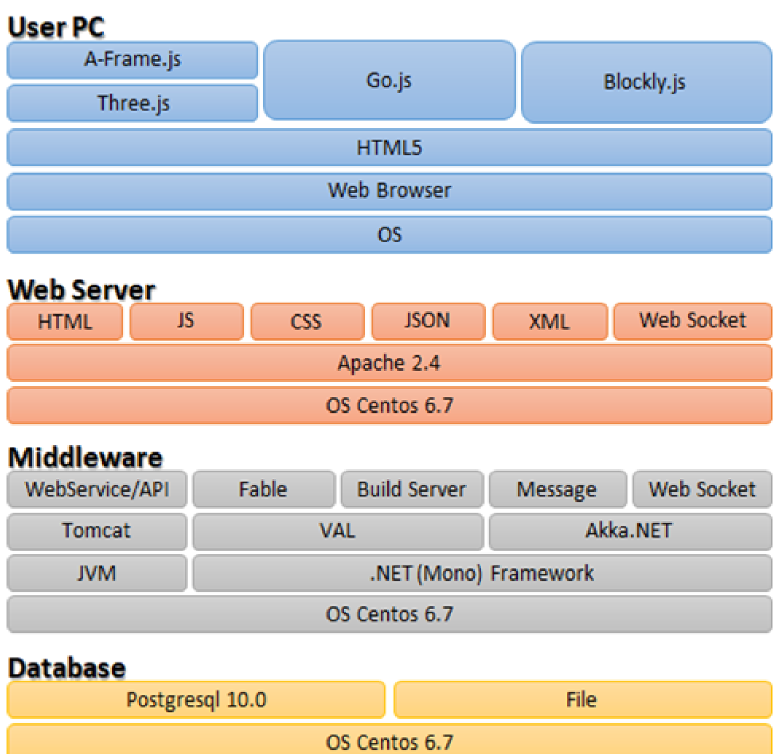

Figure 3. Proposed Platform for VR/AR Education

3.1.1. Implement 3D VR scene and object interaction in In-VR

In order to provide a sense of immersion in VR/AR, it is possible to create basic 3D objects and scenes by creating and modifying not only existing 2D screen-based editors but also HMDs and in-VR. As the goal is to construct a 3D scene that can express the planning intention of the creator in a short period of time, the object (or prefab) that has already been created is selected and placed rather than creating a realistic 3D object, will be provided.

3.1.2. Implement prototyping with Visual Language

In describing interaction, a key element of VR/AR content, Visual Language development environment is provided for those who are not professionally trained in coding education to easily accomplish their purpose. Interactions or effects that are often used in the prototyping process are provided in pre-built libraries, allowing the user to proceed with prototyping simply by selecting the appropriate interaction/effect and setting the appropriate parameters. In addition, it provides a visual language-based interaction editor that enables users to create their own interactions themselves, and makes it easy to write and test code in a virtual/augmented reality environment and easily create prototypes.

3.1.3. Linked to commercial VR engine

The prototyping platform can be used in conjunction with a commercial VR engine such as Unity/Unreal. To do this, the Visual Language code used for prototyping is converted into the script language of the target VR engine and provided. The prototyping platform can basically create 3D objects, animations, and interactions, but if you want high-quality game- and movie-like effects, you can start working in prototyping and work more efficiently.

3.1.4. VR/AR prototyping based on Functional Languages

In order to expedite the VR/AR prototyping, it provides a simple and efficient functional language with various effects and events as well as objects that can be used by the user. Functional languages are particularly strong in parallel processing, and are expected to be applied to more complex and various forms of VR/AR prototyping in the future [21][22][23][30].

The platform (see figure 4) is basically developed on the basis of Microsoft .Net Framework. VAL is selected as the language supported by Microsoft .Net Framework. Windows operating system with .Net framework is supported by various 3D engines for game, and VR/AR development is also supported most actively. Currently, .Net Framework supports other operating systems such as Linux/OS-X through Mono framework, which is an open-source project.

Actor-based AKKA.Net is used as a message infrastructure for efficient management of multiple objects in VR space. Akka.net has already begun to support the .Net framework through Akka.Net as a proven solution in Java/JVM.

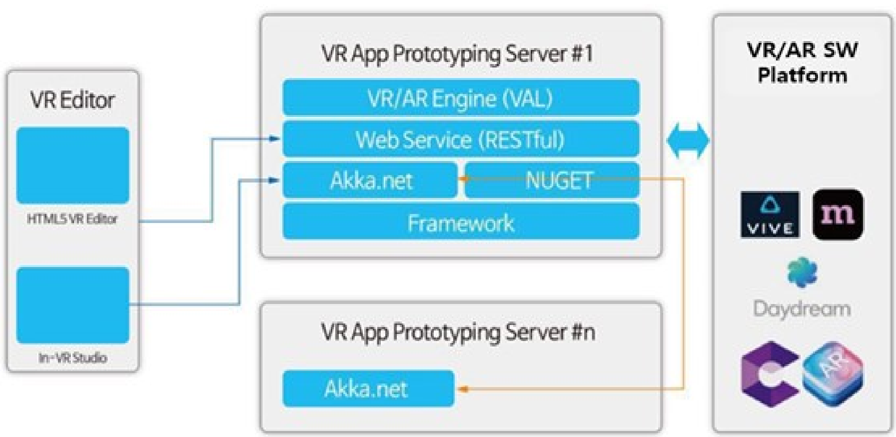

Figure 4. VR/AR App Prototyping Platform System configuration diagram

Table 1. WebVR Browser HMD Support Status

Browser / HMD	VIVE	Oculus	GearVR	Daydream	Cardboard
MS Edge	X	X	X	X	X
Firefox	O (win)	O (win)	X	X	X
Chrome(android)	X	X	X	O	O
Chromium	O (win)	O (win)	X	O	O
Samsung Internet	X	X	O	X	O

Users of this platform will mainly use the platform's main functions and services through the VR Editor. The VR Editor is divided into beginner and semi-professional users as described above, and it is recommended that the Web method is not required for the user to install and configure the software specially. Table 1 shows WebVR supported browsers that support VR with HMD.

3.2. Interface Procedure

The operation sequence of the UI(User Interface) method proposed in this paper is shown in Figure 5.

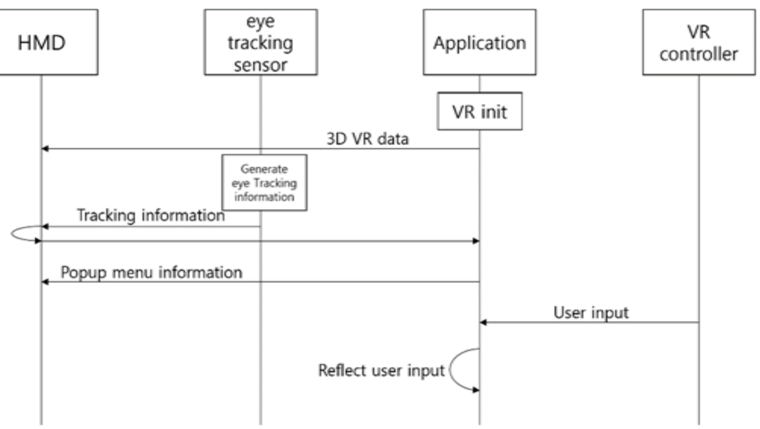

Figure 5. Procedure of the User Interface based on eye tracking device in VR environment

① The application is activated to create a VR image and deliver it to the HMD.
② The built-in eye tracking sensor grasps the area the user is interested in while watching the image displayed on the HMD, and generates the user's gaze information.
③ The eye tracking sensor transmits the created user line information to the HMD, and the HMD forwards the information to the application.
④ The application compares the user's gaze information received via the HMD with the image that is currently generated by the user, and determines which or which object the user is watching.
⑤ The application grasps the characteristics of the object and arranges menus that the object can perform.
⑥ The application combines the menu list with the VR image and delivers it to the HMD.
⑦ The HMD displays the image received.
⑧ The user chooses what he wants from the list as controller.
⑨ The application reflects the user input received through the controller and outputs a new 3D image

3.3. Framework connectivity model based on LLVM IR

In this paper, we propose a framework connection model based on LLVM IR in VR/AR development environment. As introduced in the related research above, creating an artificial intelligence development environment combined with VR/AR is a VR/AR development environment(Unity, Unreal, etc.) and a deep learning development environment(TensorFlow, NXNet, CNTK, PyTorch, Caffe2, etc.) Various front-end frameworks already exist, and various hardware(NNP, CPU, cuDNN, etc.) must also be supported. In addition, there are many challenges such as the VR/AR environment must support mobile devices or web environments.

The proposed platform applied the IR-to-IR connection method using LLVM technology. Here, the LLVM project[31] is a robust and optimized open-source compilation platform that competes with GCC in terms of compilation speed and performance of the generated code. The LLVM-based compiler consists of a part that converts from high-level source language to LLVM IR[32], and a part that provides optimization, program transformation, and static analysis. Here, the intermediate Representation(IR) refers to a data structure or code used internally to represent source code in a compiler or virtual system.

In addition, it is a WebAssembly[33] technology that can show native language-level performance in a browser. In particular, the Emscripten[30] project is an LLVM-based project that compiles native languages(C, C++) into high-performance JavaScript in the asm.js format. Currently, all commercial engines support WebAssembly. By integrating the above-mentioned contents, we propose a framework connection model as shown in Figure 6.

Centering on LLVM IR technology, it connects the existing Emscripten technology with the NNVM compiler, a new compiler in the field of deep learning integration.

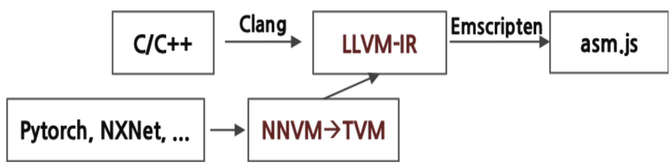

Figure 6. VR/AR framework connection model based on LLVM IR

The above model is a connection model centered on WebAssembly technology, and it is a connection model that generates LLVM IR code optimized for CUDA, OpenCL, and GPU when connected to TVM(Tensor Virtual Machine)[34].

3.4. In-VR editor using deep learning technique

Here is a simple application example implemented using the framework connection model based on LLVM IR in the VR/AR development environment. It is very difficult to design a VR environment by selecting a menu in an In-VR environment. We will introduce the process of automatically selecting and arranging the relevant assets in the process of drawing objects through real-time hand tracking and gesture recognition using the SketchRNN technique used in Google's QuickD raw Dataset.

The implementation example of Figure 7 shows how much development potential the application of artificial intelligence deep learning technology in a VR environment.

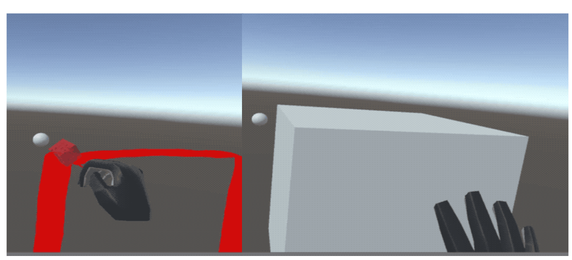

Figure 7. Automatic asset recognition and selection during the drawing process using deep learning

IV. Evaluation of the proposed system AI model

In this session, functional analysis targeting thread processing number and quantitative analysis of the proposed interface and the existing interface are conducted for the proposed system. The analysis environment assumes that the user has a single HMD, has a dedicated controller, and spends the same amount of time on communications and internal computations within the application.

4.1. Functional analysis of number of threads processed

The test was conducted in the following way to check the number of threads in the proposed platform execution environment.

① Thread target value (T): Check whether the platform execution environment creates more than 10,000 threads.
② Number of thread measurements (n): 5 times
③ Thread Test Tool: Proposed VR/AR App Platform
④ Thread measurement method: When an HTTP protocol is requested in a certain format, which is one of the functions of the proposed platform, the function of converting a specific development language (VR/AR Language) into a designated language(JS) and transmitting it is checked. If you make a request for this function in Slade, check whether the response can be received normally with JMeter.

Table 2 shows the measurement results of the number of execution threads in the proposed system environment.

Table 2. Thread measurement result

Division	1st	2nd	3rd	4th	5th
Number of threads Processed	10,000	10,000	10,000	10,000	10,000

Here, the test criteria followed ISO/IES 25023:2016 and were analyzed based on Capacity measures and User access capacity.

4.2. Quantitative analysis of user interfaces

For the interface cost analysis, we set the parameters as shown in Table 3 below. The interface consumption time(TPI) when using the conventional method can be configured as follows.

$$T_{PI}=T_{INIT}+T_L+T_{CM}+T_{MP}+T_{MS} \quad (1)$$

The proposed theory can be expressed as follows.

$$T_{NI}=T_{INIT}+T_L+T_{ET}+T_{MS} \quad (2)$$

Table 3. Consumption time parameter tables

Parameter	Description	Time(ms)
T_{INIT}	VR Operation Time	200
T_L	Time the user recognizes the object	5
T_{CM}	Time the user spent moving the cursor to the object	150-200
T_{ET}	The time at which the object is selected as an eye tracking method and the menu is displayed	10-20
T_{MP}	Time to pop up a menu by selecting an object	50-100
T_{MS}	Menu selection time	150-200

Since the T_{INIT}, T_L, and T_{MS} are the same in both schemes, the differences that occur in the two schemes actually occur in T_{CM}, T_{ET}, and T_{MP}.

Based on Equation (1), it can be seen that the time consumed by the existing UI is consumed from 555ms to 705ms at least. Based on Equation (2), it can be seen that UI method using eye tracking technique consumes from minimum 165ms to maximum 225ms.

Figure 8 ompares the consumption time between the existing controller-based UI and the eye-tracking based UI through Equation (1) and Equation (2).

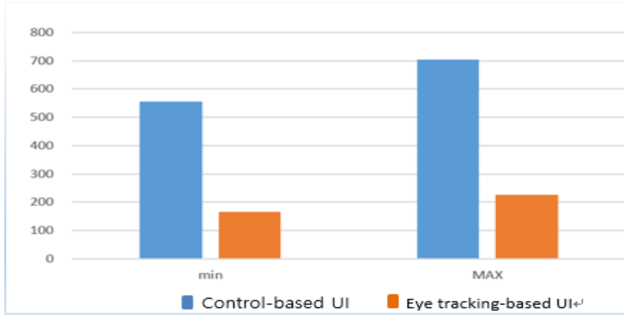

Figure 8. Consumption time comparison table

V. Conclusion

In this paper, we propose a method to use VAL more efficiently in 5G environment. The content request procedure was introduced by using the proxy server technique using edge computing technique in the existing client server model to prevent unnecessary duplicate packet requests. In the content modification procedure, a small control packet is grouped and stored in a cache, and when the control content is different from the existing content, it is driven and delivered and processed.

In addition, when a correction message for the asset comes in, the lock is turned on and the modification of other users is blocked, thereby preventing the deadlock situation.

Users who have no programming experience in this prototype platform will construct a three-dimensional scene that is the background of VR content by wearing VR HMD and arranging bundle-sets(or pre-sets) provided by the platform in a way that seems to match the Lego block in reality. And the interaction of objects in the scene(including players) is implemented by using intuitive visual language in In-VR mode, applying interaction and testing the results immediately. The user can concentrate on the more realistic scene composition and interaction implementation because the interaction creation/correction/execution-test by visual language can be proceeded without interruption with the composition/editing of 3D scene in the In-VR environment.

VI. References

[1] Wayne E. Carlson, “Section 17.1. Virtual Reality,” in A Critical History of Computer Graphics and Computer Animation: A Retrospective Overview, https://ohiostate.pressbooks.pub/graphicshistory/, Published by the Ohio State University, June 20, 2017.

[2] Chengyu Sun, Wei Hu, Diqiong Xu, “Navigation modes, operation methods, observation scales and background options in UI design for high learning performance in VR-based architectural applications,” Journal of Computational Design and Engineering., Vol. 6, No. 2, pp. 189-196, April 2019.

[3] Hao Song, Qingjin Peng, Jian Zhang, Peihua Gu, “A VR-based Interactive System to Improve Users’ Involvement in OAP Development,” Procedia CIRP., Vol. 56, pp. 13-18, 2016.

[4] R. Pagés, K. Amplianitis, D. Monaghan, J. Ondřej, A. Smolić, “Affordable content creation for free-viewpoint video and VR/AR applications,” Journal of Visual Communication and Image Representation., Vol. 53, pp. 192-201, 2018.

[5] Philipp A. Rauschnabel, Reto Felix, Chris Hinsch, “Augmented reality marketing: How mobile AR-apps can improve brands through inspiration,” Journal of Retailing and Consumer Services., Vol. 49, pp. 43-53, 2019.

[6] Abdullah, Dakhaz Mustafa, and Siddeeq Y. Ameen, “Enhanced mobile broadband (EMBB): A review,” Journal of Information Technology and Informatics., Vol. 1, No 1, pp. 13-19, 2021.

[7] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Introduction to Ultra Reliable and Low Latency Communications in 5G," arXiv preprint arXiv:1704.05565, 2017.

[8] Bockelmann, Carsten, et al., "Massive machine-type communications in 5G: Physical and MAC-layer solutions," IEEE communications magazine Vol. 54, Issue 9, pp. 59-65, 2016.

[9] Hewitt D. Crane and Carroll M. Steele, "Generation-V dual-Purkinje-image eyetracker," Applied Optics. Vol. 24, Issue 4, pp. 527-537, 1985.

[10] Md Fasiul Alam, Serafeim Katsikas, Olga Beltramello, Stathes Hadjiefthymiades, "Augmented and virtual reality based monitoring and safety system: A prototype IoT platform," Journal of Network and Computer Applications, Vol. 89, 1, pp. 109-119, 2017.

[11] Zhao, Jianzhou, et al., "Formalizing the LLVM intermediate representation for verified program transformations," in Proc. of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, 2012, pp. 427-440.

[12] Georgios Mylonas, Christos Triantafyllis, Dimitrios Amaxilatis, "An Augmented Reality Prototype for supporting IoT-based Educational Activities for Energy-efficient School Buildings," Electronic Notes in Theoretical Computer Science, Vol. 343, pp. 89-101, 2019.

[13] Scott G. Dacko, "Enabling smart retail settings via mobile augmented reality shopping apps," Technological Forecasting and Social Change, Vol. 124, pp. 243-256, 2017.

[14] L. Gallo, A. Minutolo, G. De Pietro, "A user interface for VR-ready 3D medical imaging by off-the-shelf input devices," Computers in Biology and Medicine, Vol. 40, Issue 3, pp. 350-358, 2010.

[15] Zhou, Shuchen, and Waqas Jadoon, "The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment," Computer Networks, Vol. 178, 107334, 2020.

[16] Honghe Huang, Yi Xu, Xiao Hua, Weixiong Yan, Yanjie Huang, "A crowdsourced system for robust eye tracking," Journal of Visual Communication and Image Representation, Vol. 60, pp. 28-32, 2019.

[17] David A. Robinson, "A method of measuring eye movement using a scleral search coil in a magnetic field," IEEE Transactions on Bio-Medical Electronics, Vol. 10, Issue 4, pp.137–145, 1963.

[18] Yuka Yasui, Junko Tanaka, Masaki Kakudo, Masahiro Tanaka, "Relationship between preference and gaze in modified food using eye tracker," Journal of Prosthodontic Research, Vol. 63, Issue 2, pp. 210-215, 2019.

[19] Dan Witzner Hansen, Qiang Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, Issue 3, pp. 478-500, 2010.

[20] Mana Farshid, Jeannette Paschen, Theresa Eriksson, Jan Kietzmann, "Go boldly!: Explore augmented reality (AR), virtual reality (VR), and mixed reality (MR) for business," Business Horizons, Vol. 61, Issue 5, pp. 657-663, 2018.

[21] Cameron Wakefield, Henk-Evert Sonder, Wei Meng Lee, "The Microsoft .NET Framework," in VB.net Developer's Guide, 2001, pp. 33-89.

[22] Fabian Boemer, Yixing Lao, Rosario Cammarota, and Casimir Wierzynski, "nGraph-HE: A Graph Compiler for Deep Learning on Homomorphically Encrypted Data," in Proc. of the 16th ACM International Conference on Computing Frontiers, 2019, pp. 3-13.

[23] Scott Cyphers, Arjun K. Bansal, Anahita Bhiwandiwalla, Jayaram Bobba, Matthew Brookhart, Avijit Chakraborty, Will Constable, Christian Convey, Leona Cook, Omar Kanawi, Robert Kimball, Jason Knight, Nikolay Korovaiko, Varun Kumar, Yixing Lao, Christopher R. Lishka, Jaikrishnan Menon, Jennifer Myers, Sandeep Aswath Narayana, Adam Procter, Tristan J. Webb, "Intel nGraph: An Intermediate Representation, Compiler, and Executor for Deep Learning," 2018, ArXiv:1801.08058.

[24] Matic Perovšek, Janez Kranjc, Tomaž Erjavec, Bojan Cestnik, Nada Lavrač, "TextFlows: A visual programming platform for text mining and natural language processing," Science of Computer Programming, Vol. 121, pp. 128-152, 2016.

[25] Baltasar García Perez-Schofield, Francisco Ortin, "A didactic object-oriented, prototype-based visual programming environment," Science of Computer Programming, Vol. 176, pp. 1-13, 2019.

[26] Barsoum. E, Kuester. F, "WebVR: An interactive web browser for virtual environments,", In Stereoscopic Displays and Virtual Reality Systems, Vol. 5664, pp. 540-547, 2005.

[27] J. Y. Xia, B. J. Xiao, Dan Li, K. R. Wang, "Improvement of virtual EAST system based on WebVR," Fusion Engineering and Design, Vol. 127, pp. 267-274, 2018.

[28] Wang Chun, Li Lei, Yuan Jiwu, Zhai Liangyun, Liu Gang, "Development of Emergency Drills System for Petrochemical Plants Based on WebVR," Procedia Environmental Sciences, Vol. 10, Part A, pp. 313-318, 2011.

[29] Dongwei Liu, Jie Peng, Yanyu Wang, Meijin Huang, Qianshan He, Yafei Yan, Bingxin Ma, Caijun Yue, “Implementation of interactive three-dimensional visualization of air pollutants using WebGL,” Environmental Modelling & Software, Vol. 114, pp. 188-194, 2019

[30] Zakai, Alon, “Emscripten: an LLVM-to-JavaScript compiler,” in Proc. of the ACM international conference companion on Object oriented programming systems languages and applications companion(OOPSLA '11), 2011, pp. 301-312.

[31] C. Lattner and V. Adve., “LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation,” in Proc. of the International Symposium on Code Generation and Optimization 2004(CGO 2004), 2004, pp. 75-86.

[32] llvm.org., LLVM Language Reference Manual, URL: http://releases.llvm.org/6.0.0/docs/LangRef.html, 2018.

[33] HAAS, Andreas, et al., “Bringing the web up to speed with WebAssembly,” in Proc. of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation(PLDI 2017), 2017, pp. 185-200.

[34] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Meghan Cowan, Haichen Shen, Leyuan Wang, Yuwei Hu, Luis Ceze, Carlos Guestrin, Arvind Krishnamurthy, “TVM: An Automated End-to-End Optimizing Compiler for Deep Learning,” in Proc. of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), 2018, pp. 578-594.

Authors

Hyun-Sook Lee

2021.02 : Ph.D. of Industrial Systems Engineering, Dongguk University
2023.03 ~ Present: Research Professor in Dept. of AI.SW Convergence Research Institute, Kangnam University

Research Interests: Data Science, Intelligence SW, VR/AR Education Platform Metaverse Platform, etc

Jee-Uk Heu

2016.02 : Ph.D. of Computer Science and Engineering, Hanyang University
2021.03 ~ Present : Assistant Professor in Dept. of KNU Cham-Injae College, Kangnam University

Research Interests : Multi-Media Retrival, SNS analysis, Big-data, Multi-Document Summarization, Artificial Intelligence, Natural Language Processing, LLMS, etc