Implementation of a Quality Measurement Software for Virtual Reality Content on a head mount display

Ko Takayama
Graduate School of Engineering
Fukuoka Institute of Technology
Fukuoka, Japan
mgm18102@bene.fit.ac.jp

Yushi Machidori
Graduate School of Engineering
Fukuoka Institute of Technology
Fukuoka, Japan
mgm19106@bene.fit.ac.jp

Kaoru Sugita
Dept. of Information and
Communication Engineering
Fukuoka Institute of Technology
Fukuoka, Japan
sugita@fit.ac.jp

***Abstract*—In a VR streaming content, an application software is necessary to control both a content quality on a streaming service and a display quality on a head mount display (HMD) according to loads for head mount display, computer facilities and network considering a user's requests. In this paper, we focus about a quality measurement software for Virtual Reality (VR) content to control display quality on a HMD and discussabout its implementation.**

Keywords—VR, QoS, Quality Measurement, HMD

I. Introduction

Nowadays, the Virtual Reality (VR) has become a popular technology for general people caused by low cost VR devices. Many users can play VR contents such as VR videos and VR games over the Internet and can have higher immersive experiences in a virtual space. These VR contents are used for a stereo image to give a parallax feel sense of 3D expression (IE1) and an omnidirectional viewing synchronized with head direction (IE2) as shown in Fig. 1. However, it is difficult to keep playing the VR contents because of limitations on the network performances during rush hours on the Internet. For this reason, the VR content should keep the QoS (Quality of Service) and the QoS parameters should be changed simultaneously without being noticed by the user [1].

There are some studies proposed for streaming of virtual reality content to mobile users. In [2], the VR content can be reduced when there is a deterioration of quality of the content which is played at high bit rate for Region of Intensity (ROI) and at a low bit rate for other regions in an immersive omnidirectional content provided to control a viewpoint. In [3], a framework is introduced for a resource allocation to construct a VR model and optimize QoS parameters in the VR contents. However, these studies have not investigated the effects of immersive experience in streaming contents.

In our work, we focused on the streaming framework of VR contents for keeping the immersive experience reflected by QoS parameters in the Internet environment. We have already introduced a QoS management framework for VR contents to gives priorities and change the qualities according to the limitation of available resources and the user's requests [4]. We also have implemented a VR streaming software [5] and proposed a framework of QoS control for VR content [6]. In these works, we present the implementation of a VR streaming software to find the appropriate reduction of the data size for QoS parameters in different types of video formats. However, we have discussed about qualities on a streaming content but not qualities on the HMD. In this paper, we show an implementation of a window MR application for obtaining quality parameters of playing qualities for the HMD.

This paper is organized as follows. The quality measurement software for VR content is overviewed in Section II. The implementation are presented in Section III. Finally, Section IV concludes the paper.

II. A Quality Measurement Software for Virtual Reality Content

Our quality measurement software is organized as a client-server system and can measure following quality parameters on the HMD during playing a VR content as shown in Fig.2.

(MQ1) IPD (Inter-pupillary distance: the distance between the user's eyes),

(MQ2) The field of view of the camera in degrees,

(MQ3) The viewport's aspect ratio,

(MQ4) The total number of frames that have passed,

(MQ5) The Refresh rate of the display in Hertz,

(MQ6) The current height and width of an eye texture for the loaded device,

(MQ7) The frame rate of the clip or URL in frames/second,

(MQ8) The width of the images on the video clip, or URL, in pixels.

In the software, the client displays an omnidirectional video and measures video qualities on the HMD. The server records a measurement data received from the client.

III. Implementation

The client software implemented as a Windows MR application on a Window MR HMD and a Windows 10 system as shown in Table I and Table II, respectively. The server software implemented as a console application on a Mac mini as shown in Table III.

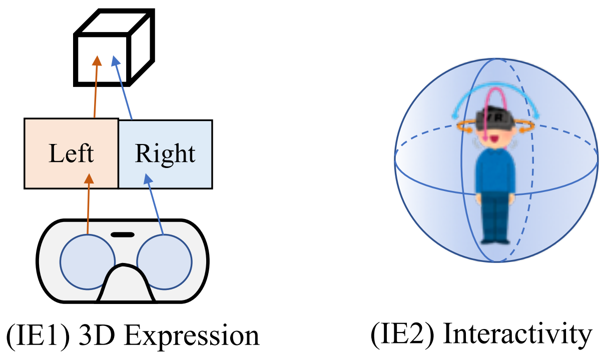

Fig. 1. Important factors of VR content.

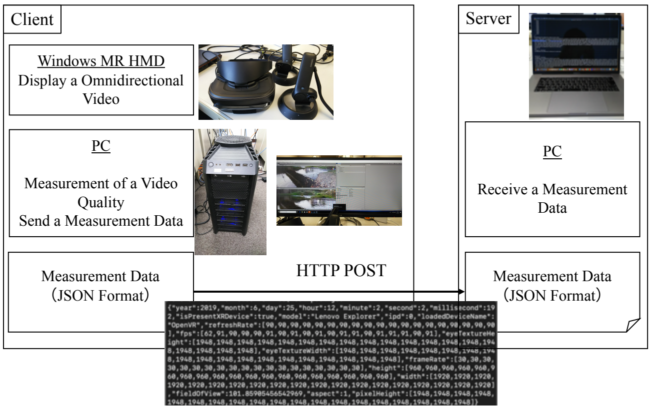

Fig. 2. Organization of Quality Measurement Software for a Virtual Reality Content.

TABLE I. SPECIFICATION OF HEAD MOUNT DISPLAY

Display Size	2.89 [inch]
Resolution	2880 x 1440 [pixel] (1440 x 1440 [pixel] for an eye)
Viewing Angle	110 [Degree]
Refresh Rate	90 [Hz]

TABLE II. CLIENT PC

CPU	Intel Core i5 9400F
RAM	DDR4 PC4-21300 32GB
OS	Windows 10 Education
GPU	NVIDIA GeForce RTX 2060

TABLE III. SERVER PC

CPU	Intel Core i7 2.6 GHz
RAM	DDR4 PC4-19200 16GB
OS	macOS Mojave version 10.14.5

The Windows MR application is composed of 6 objects instantiated from classes as shown in Fig.3 – Fig.8. These classes are developed by C# Language using Unity, Open VR and a scripting API on the Unity as shown in Table IV. Especially, The QoS parameters on the Windows MR HMD are obtained by the Unity APIs as shown in Table V. Each class supports the following functions.

(O1) MeasureMain class instantiates objects of the application and measures quality parameters on the application,

(O2) Vrparameters class sets measurement condition for the application and stores results,

(O3) TimerCounter class releases a message driven a measurement event for quality parameters at specified interval on the application,

(O4) FpsConfigration class initializes a frame rate on the application,

(O5) IPD class acquires an inter-pupillary distance,

(O6) SendJson class posts measurement results as binary data to the server software using HTTP.

A sequence diagram is in a measurement of quality parameters as shown in Fig. 9. Firstly, the MaesureMain object instantiates VrParameters and TimerCounter objects. Next, the MaesureMain object sets a frame rate of the application to FpsConfiguration and gets the IPD. After that, the MaesureMain object measures quality parameters on the application and send them to the server software. Examples of measurement results are JSON format data as shown in Fig.10 and Fig.11. These results were taken during playing a VR content for a normal quality and a low quality on the application.

IV. CONCLUSIONS

In this paper, we describe a quality measurement software for VR content on the HMD and its implementation. The software can monitor quality parameters on the HMD during playing a VR content at the client software and store these data at the server software. Currently, we are evaluating an accuracy of measurement results on our application, implementing a video streaming module to control a QoS parameters for a VR content playing on the Windows MR HMD. In the future, we will conduct a quality of experience parameters for a VR content in the experimental way.

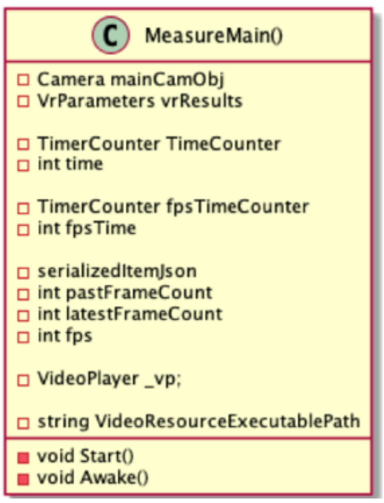

Fig. 3. MeasureMain Class

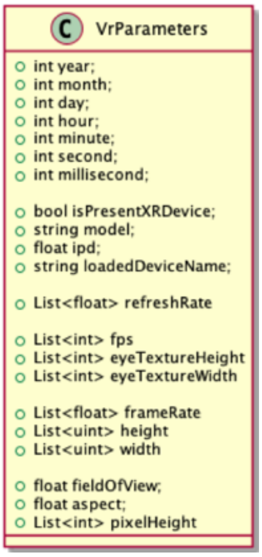

Fig. 4. VrParameters Class

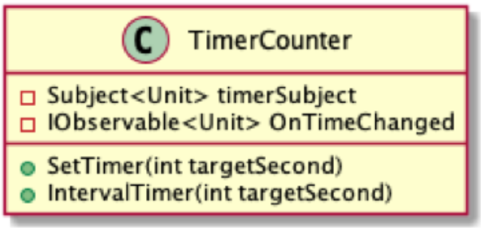

Fig. 5. TimerCounter Class

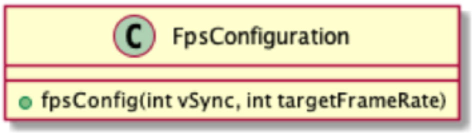

Fig. 6. FpsConfiguration Class

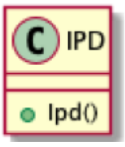

Fig. 7. IPD Class

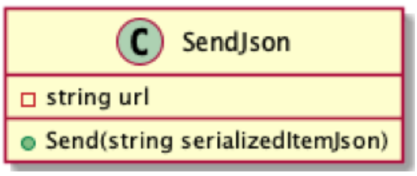

Fig. 8. Sendjson Class

TABLE IV. DEVELOPMENT ENVIRONMENT

Game Engine	Unity 2019.1.0f2 - preferred Personal
Build Settings	Windows Standalone
Virtual Reality SDKs	OpenVR
Scripting API	UnityEngine.Video.VideoPlayer

TABLE V. UNITY API TO MEASURE A QUALITY OF VIRTUAL REALITY CONTENT

Unity API	Measurement of Data
InputTracking.GetLocalPosition(XRNode.LeftEye) InputTracking.GetLocalPosition(XRNode.RightEye)	IPD (The distance between the user's eyes will affect the position of the position of the node in its local tracking space)
XR.XRDevice.isPresent	A VR device in working order
XRDevice.model	A model of loaded VR device
XRSettings.loadedDeviceName	Type of VR device that is currently loaded
Camera.fieldOfView	The field of view of the camera in degrees
Camera.aspect	The viewport's aspect ratio
Time.FrameCount	The total number of frames that have passed
XRDevice.refreshRate	The Refresh rate of the display in Hertz
XRSettings.eyeTextureHeight XRSettings.eyeTextureWidth	The current height and width of an eye texture for the loaded device (Maximized value: 1948)
VideoPlayer.frameRate	The frame rate of the clip or URL in frames/second
VideoPalyer.width / VideoPlayer.height	The width of the images in the video clip, or URL, in pixels

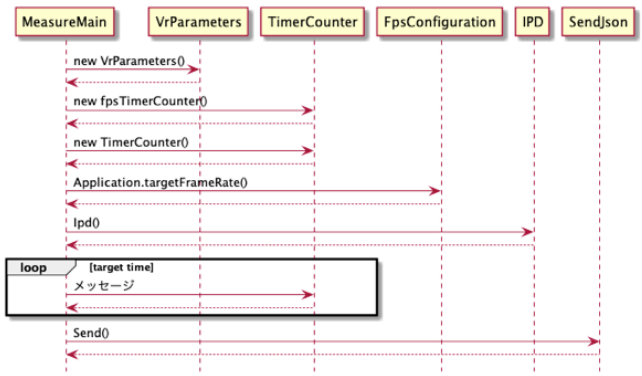

Fig. 9. A Sequence Diagram in a Measurement of Quality Parameters

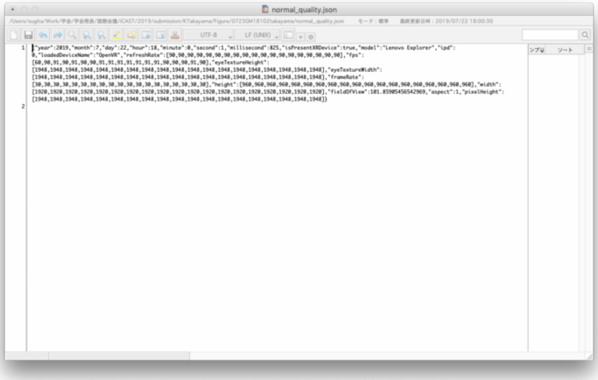

Fig. 10. A Measurement Result for a Normal Quality Content

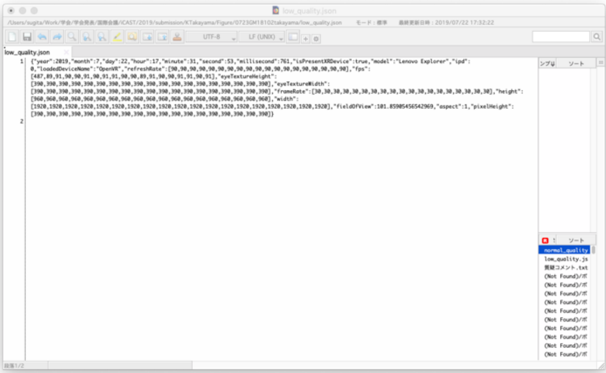

Fig. 11 A Measurement Result for a Low Quality Content

References

[1] B.Ejder, B.Mehdi, M.Muriel, D.Mérouane, "Toward Interconnected Virtual Reality: Opportunities, Challenges, and Enablers," IEEE Communications Magazine, Vol. 55, Issue. 6, pp.110-117, 2017.

[2] A.Hamed, E.Omar, H.Mohamed, "Adaptive Multicast Streaming of Virtual Reality Content to Mobile Users," Proceedings of the on Thematic Workshops of ACM Multimedia, pp.170-178, 2017.

[3] C.Mingzhe, S.Walid, Y.Changchuan, "Resource Management for Wireless Virtual Reality: Machine Learning Meets Multi-Attribute Utility," IEEE Global Communications Conference, pp.4-8, 2017.

[4] K.Takayama, K.Sugita, "QoS Management for Virtual Reality Contents," Proc. of International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2018), Advances in Intelligent Systems and Computing, Springer, Vol.773, pp.329-335, 2018.

[5] K.Takayama, Y.Machidori, K.Sugita, "Implementation of a Virtual Reality Streaming Software for Network Performance Evaluation," Proc. of International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2018), pp.381-386, 2018.

[6] K.Takayama, K.Sugita, "Proposal for a framework of QoS Control for Virtual Reality Contents", DPS Workshop-2018, IPSJ, pp.117-122, 2018 (in Japanese).