Online Open Campus Using Remote AR and Avatar for Real Time Communication

Hao CHEN
44201034-5

Master of Engineering

Supervisor: Prof. Jiro Tanaka
Graduate School of Information, Production and Systems
Waseda University

January 2022

Abstract

Due to COVID-19, online open campuses have become popular in many universities and schools. However, the lack of interaction with environment and intuitive user representation in campus result in poor communication between users.

This research aims to improve the communication of online open campuses. We propose and implement a method of using remote AR for online open campuses to allow users to visit the campus and use AR functions remotely. The system can recognize the environment of campus with remote AR technologies. For better communication, users can point out the objects intuitively with AR pointer and control their avatar to interact with the physical world.

There are two cases in our user case, group visiting for general public area and laboratory visiting for room-scale area. Corresponding to this, our proposed system consists of two approaches.

The core idea is the video-based AR technique used in the first approach. Video-based AR is an innovative method of capturing environment and AR Session information by saving intermediate data in the form of video files. It solves the limitation in general AR systems that users have to capture the environments on-site by themselves and the system transmits intermediate data in real-time.

Another approach is using 3D model generated by LiDAR. It also allows user visit the campus remotely and using the same sharing functions. It was used as a supplement in the different cases to the first approach, and we can also integrated them in a good ways.

With the capability of remote AR, we introduced avatar into the system to help users to communicate with others. The sharing functions are based on avatar's action and can be used in both two modes.

Keywords: Augmented Reality, Video-Based AR, 3D Model, Avatar, Communication

Acknowledgements

First of all, I would like to thank my supervisor Prof. Jiro Tanaka, for accepting me to join IPLAB as the beginning of this thesis. For the past two years, I have been honored to got his patient guidance in academy. From lectures to discussing topics to reviewing thesis, his help and suggestion were indispensable at every stage of this work. In addition to his direct help, his rigorous attitude to research and the kindness to people also serve as an excellent example of researcher that inspire me all the time.

Secondly, I would like to thank all IPLAB members. You have accompanied me through a very pleasant life in my master course, especially during these special time of COVID-19. Whether it's the happy times of progressing together or the times of overcoming difficulties together, it's very meaningful to me.

Finally, I would like to express my gratitude to my family. They have always supported me financially and spiritually. They provided me with a great environment for study at home during M1 of being unable to come to school due to COVID-19. After I came to Japan, they still always cared and encouraged me.

Contents

List of figures

Chapter 1

Introduction

1.1 Background

Open campus is traditionally an offline event held by the school to promote the campus to the candidate students. However, due to COVID-19, online open campus has become more popular. Now video conference (e.g. Zoom [1]) is widely used as a simple solution, and some prior virtual tour systems can also be considered as a possibility. These methods can meet the basic requirements of remote visit, but considering that students need to communicate to obtain school information, they all have limitation in the support of communication. The problem is that remote users are completely separated from the environment they visit, resulting in users not being able to see their peers in the environment and lacking an intuitive means of using the environment's information in a conversation.

As we know, video files can take time as the axis to save 2D visual history information. On the other hand, with the development of smartphones, frameworks such as ARcore allow the use of AR through smartphone. It usually require a real-time stream from the smartphone camera. But in a new feature recently released by ARCore [2], it indicated that supported Android phones are allowed to record 3D information of the scene and motion information of the device in the same way as video files record 2D visual information, which can be used to record and restore the AR Session.

Augmented Reality (AR) can make the system recognize the environment in 3D and superpose virtual objects in the real world. Some typical AR systems might use plane detection to interact with the environment[3, 4] or virtual avatars to aid communication with the user[5, 6]. Therefore, we consider that using the above technologies to share AR Session combined with real-time synchronization of user interaction information such as avatar and pointer, can be used as a new form of remote AR to improve communication in remote open campuses.

1.2 Organization of the Thesis

This thesis is organized with 6 chapters: Chapter 1 introduce the background of this research. Chapter 2 first introduces the goals of this research and the approaches we propose, and then analyzing the scenarios of the online open campus we give two typical use cases and corresponding approach at the end. Chapter 3 is organized according to the techniques, make some introductions to recent related work, and analyze the difference with this research. Chapter 4 will give the talk of the system design, including the overview, the introduction of the core idea video-based AR and others design to improve the communication and user experience. Chapter 5 is about the implementation of the system. We organize the sections by function. Chapter 6 finally gives the conclusion of this thesis and our future work.

Chapter 2

Research Goal and Approach

2.1 Goal

The goal of this research is to provide an online open campus system based on remote AR technology, which uses AR annotations and Avatar to provide real-time communication.

Existing online open campuses based on video conferencing can only provide users with 2D visual information, and the communication between users is limited to sharing annotations and text on the screen. As we know, Augmented Reality (AR) has the abilities of spatial awareness and superposing virtual objects of, therefore has potential for the communication with spatial context.

However, general AR systems have the limitation that users have to capture the environment on-site by themselves. Remote AR [7–10] further provides a way for remote users to watch the video captured from host side and interact with the environment using AR. In some previous systems, remote users would wear VR HMDs[7]. But recently with the development of mobile devices, it has also become possible for remote users to use mobile phones or tablet[10]. Considering the users of the open campus, using mobile phones by remote users can make the system more convenient. In addition, in the scenario of an open campus, we think that we should provide users with different degrees of freedom of operation, such as group mode and individual mode, depending on the scene of the visit.

Avatars[11, 12] are usually full-body, half-body or face-only virtual humanoid character that are used to represent users in VR/AR systems. Avatars have body language and expressions, and by controlling these actions of avatars, we can improve the communication between users.

As a summary, the goal can be divided into the following several steps:

1. Provide the method of campus visit based on remote AR;
2. Provide different operating freedom modes based on scenes;
3. Provide the method of using Avatar for multi-user real-time communicate.

2.2 Approach

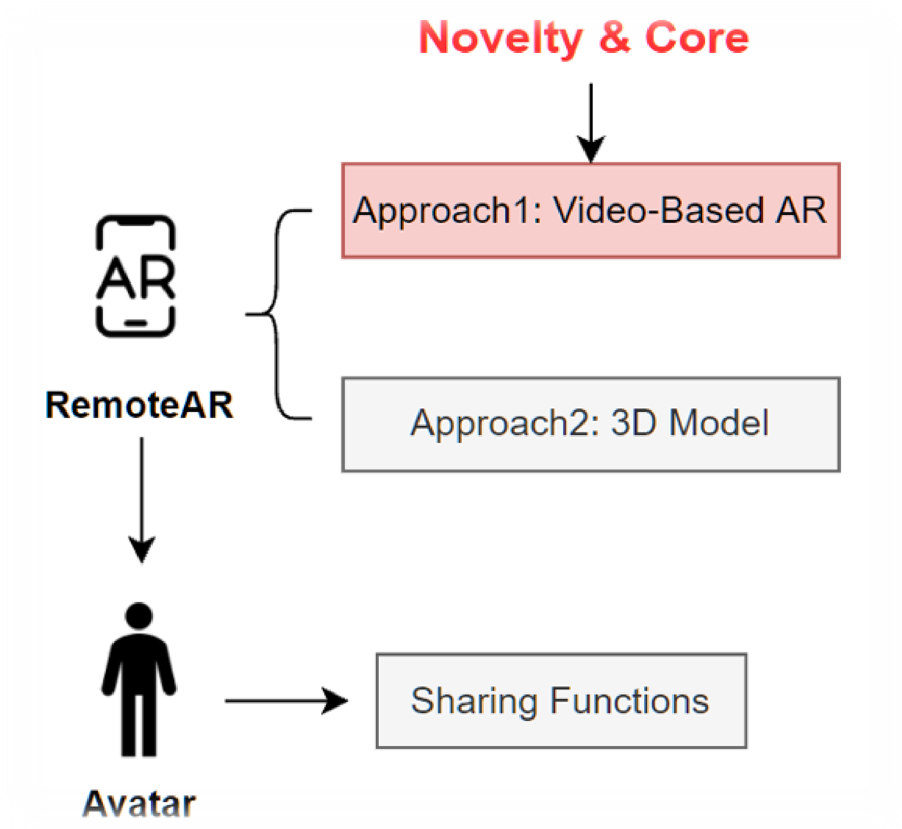


Fig. 2.1 Two Approaches for RemoteAR and Functions based on Avatar

Fig.2.1 shows the overview of our proposed approaches. We use the Video-Based AR technique and 3D Model for remote AR in difference visited scenes and design Sharing Functions based on Avatar for real-time communication.

Video-based AR is the novelty and core idea of remote AR approaches. ARCore provides the recording and playback APIs for video-based AR [2]. Using the recording API to capture a video for AR session, the system will store a data set containing the video stream, depth map, and motion data. The playback API uses the data set to reconstruct the AR session as playing the video. It can be noticed that there are no time and location restrictions when using AR through the playback function. It allows remote users to use the AR function in the video of a specific location, so we propose that multiple users can access the same remote AR session by ensuring that users playback at the same time.

Another approach is 3D model. In previous works, 3D models were used by remote users in remote AR collaboration systems for viewing and performing AR functions [13, 7, 14]. We scan the 3D models of campus with LiDAR. In a room-scale area such as laboratory rooms, we can get the model of room in a accepted quality. By controlling the avatar to explore the model, user can access the place remotely for more detailed observation compared to the video-based AR approach. In the video mode, 3D models also can be used as the navigation map.

Sharing functions are functions based on avatar's action or AR annotations for multi-user communication. They mainly synchronize user interaction data and use AR features for display or processing, such as using AR pointer and ray for object pointing and AR panels for user profile.

2.3 Use Case

In this section, we will introduce the use case we assumed. A typical scenario is considered as that: a student attend the online open campus using smartphone at home. Around the appointed time, the student enter the waiting room and wait for the other students to set off together. When the visiting begins, the student can see the campus scene and other group members appear on campus as avatars. The system guides all students according to the route, starting from the entrance to the student lobby, and then to the 2nd floor. When visiting the student lobby, the student notice a interesting poster. The student can point out it with screen touch and ask who has the information about this poster. Other students can understand which poster is being talked about from the avatar's action and the AR pointer on the poster. When coming to 2nd floor, the student would like to choose the most concerned laboratory to visit. So, the student can click the door from the corridor to enter the laboratory room. In the laboratory room, the system will not lead a route. But the student can move by himself/herself to observe the object in laboratory in more detail. If the student found some place of interest, the same operation can be used to point out the object.

Referring to the use case, we found that there are generally two modes of visits. One is a group visit organized in a public space, and the other is a free movement visit in a more private space, we assume it is in a laboratory.

Through the interaction with the door, the user can switch between different scenes and can switch to different modes according to the different target scenes.

2.3.1 Group Visiting

We apply the video-based AR to the group visiting. In this case, users will gather together and start to visit at the same time, and can see other users and use the sharing functions to communicate. It is mainly to visit while walking, the speed of movement depends on the video recorder, and the route of the visit is also determined in advance during recording. Because we consider that most of the visitors to the open campus are here for the first time, it is difficult to ask visitors to choose their own routes in unfamiliar places. We think this may also be one of the reasons why traditional video tours are popular in open campuses.

2.3.2 Laboratory Visiting

Group visiting is always on the move. However, in a room-scale space, there is another case where the user prefer to standby and observe carefully. Therefore, laboratory visiting case serves as a supplement to the group visiting. We apply the 3D model mode to this case. User can control the avatar walk around and communicate with the users in the same room. The sharing function for communication can also be used in both modes.

Chapter 3

Related Work

3.1 Remote Augmented Reality

Remote AR generally uses the spatial recognition ability of the local AR device to transmit visual information and motion data to remote devices. Comparatively the remote device reconstructs the environment information by processing the data and uses it to simulate the native AR session. In both sides, the users can superimpose digital information on the same place with the native or simulated AR session. And usually there is a separated thread or service will synchronize the interactive information between remote users and local users through the network. One of the challenges is what kind of intermediate data to be used to transmit visual and motion information. Here we have to consider balancing the user experience and transmission bandwidth.

Using video streams for visual information is a common approach[15–18], but because the video is flat and lacks 3D information, it is difficult for the system to provide complex AR interactions. Sometimes even though 3D representation is used, the 3D information of the environment is not used when calculating the position of virtual objects.

In the works of Kim et al.[16, 19], the local user wears a AR HMD that shares live video of local environment to another remote user who wears a VR HMD. And users share a cube as the pointer to communicate with each other. The result of its user study shows that there are still difficulties in drawing 3D sketches at a certain depth.

In order to support more complex interaction with environment in higher accuracy, we want to use the depth information and spatial information such surface and plane by AR technique. Now many smartphones have been developed with AR capabilities. They are equipped with depth camera and motion sensor that enable the plane detection and camera tracking in AR. Compared to HMDs, mobile devices are more convenient.

In the meantime, many systems of remote AR have been proposed to improve the user experience by sharing virtual hands, user avatars and pointer [11, 17, 9, 20]. In these systems, adapting the virtual contents to the real environment's is a challenge. Mini-Me[11] is a remote AR collaboration system with the avatar that can adapt with the environment by redirected gaze and gestures. Similarly, in the subsequent chapters of system design, we also made adaptive designs for the avatar's position alignment and the redirection of pointing actions in order to improve the user experience.

3.2 Virtual Tour

As a possibility of online open campus, virtual tour is widely used because there are developed commercial products that can be utilized immediately. At present, many universities adopt some video conferencing software (e.g. Zoom). Through the video conference software, real-time or pre-recorded video streams can be shared by a local host to remote users and voice chats can be conducted between users to meet basic visit and communication requirements. The advantage of video is that it can truly restore the details of the display scene. However, the shortcomings of video conference is are also obvious, the video is mainly for watching, while how to communicate with other users while watching is still preliminary.

Boffi et al.[21] proposed a virtual driving tour system allows users to wear VR HMD to watch real-time 360 videos of a remote car. It lacks multi-user support and communication between users. Nassani et al.[15] extended an open-source video conferencing platform for virtual tours with interactions. Users can control the perspective in a 360-degree video sharing meeting and move their pointers to be placed on the object they are interest in. While the pointer can only be placed on a fixed curved surface relative to the front of the field of view.

In the online open campus application, in order to solve the communication problem, we use AR technology to display the user's avatar and the user's pointer in the video to improve communication between users.

Alternatively, different from the video conference, Li et al.[22] used 3D model for the virtual tour of a campus and Priolo et al.[23] used 3D models for the virtual tour of a museum. Sanker and Seitz [24] further combined motion capture in a virtual campus tour. Compared to video-based virtual tours, these systems can provide more freedom of visit details. But for large scenes, it is difficult to improve the quality of the model to achieve the same realism as the video.

In order to take advantage of different methods, our system combines video and 3D model methods. For large spaces, we use video methods and for small spaces, we use 3D model-based methods. The user can switch between two modes when switching scenes on the move.

3.3 3D Model Reconstruction

Using a depth camera can scan an object or environment to generate a 3D model[13, 25]. Recently, scanning with LiDAR can improve the accuracy of depth map to get the model with higher quality [26, 10].

Sankar et al.[27] proposed a method using AR user interface with mobile devices to capture the room model in an interactive way. It shows the usability of the spatial awareness of AR that provides high-level environment information such as planes and object in the 3D models.

Tecchia, Alem, and Huang[9] used the Kinect to capture the 3D meshes in real-time for remote users in a remote AR collaboration system. Since the 3D meshes are continuously updated in real-time, it require high bandwidth and users always can only see incomplete scenes. Wang et al.[10] captured a full 3D model one time with LiDAR and used it for remote

AR authoring. Teo et al.[7] further combined the static modes with 360-degree live video in remote AR collaboration. It reduces bandwidth requirements and also provides a certain amount of AR spatial recognition capabilities. However, because the static model and the 360 live video are from different times, the two different environments may not match when objects changed.

In the 3D model based remote AR systems, consider how user data should be represented in addition to the environment model. Gao et al.[8] explored using point-cloud to represent the remote user's hands. It achieved a minimal representation, which is sometimes harder for the user to recognize compared to the 3D AR objects.

Chapter 4

System Design

4.1 System Overview

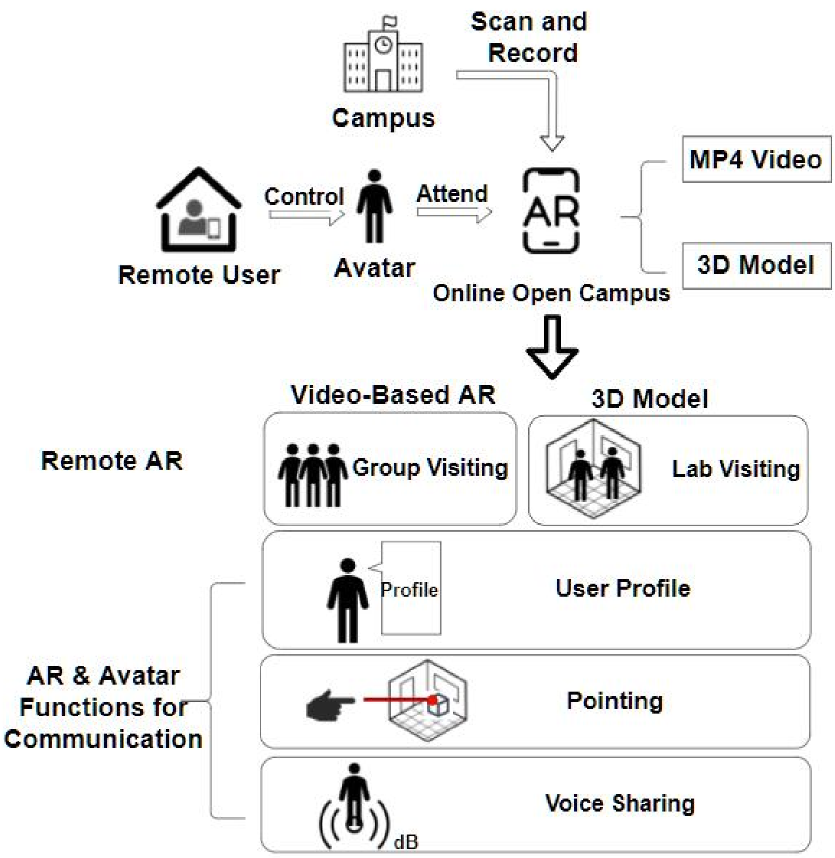


Fig. 4.1 Overview of our approach

Fig.4.1 shows the overall structure of the system. We will firstly introduce the techniques of remote AR and how we use these techniques in the system for remote visiting. Then introduce the sharing functions based on AR and avatars for communication. Finally, we will introduce the other functions for improving the user experience.

(a) Pixel4 and Pixel5

(b) iPad Pro

Fig. 4.2 Hardware

Fig.4.2 shows the hardware of the system. We use Google Pixel4 and Google Pixel5 as the clients. For each user, an Android phone which supports ARCore framework is required. And user also needs the network to communicate with the server. We use an iPad Pro equipped with LiDAR camera to generate the 3D models of the room.

4.2 Video-Based AR

The video-based AR technique [2] is the main approach we use for remote AR. Because the video-based AR can remove the location and time constraints from general AR.

General AR relies on the visual information from the real-time camera and motion information from sensors. Because they are all real-time data and the system render AR environment in real-time, user can only use general AR in the local place. If take the mobile AR as an example, the depth camera provides the depth map that can be used to analyze the environment of a frame. The IMU(Inertial Measurement Unit) of the mobile phone usually includes an accelerometer and a gyroscope, which provide pose information of consecutive frames, so as to realize the recognition and digitization of the entire scene. In addition, the RGB camera provides users with the real-world appearance required for interaction.

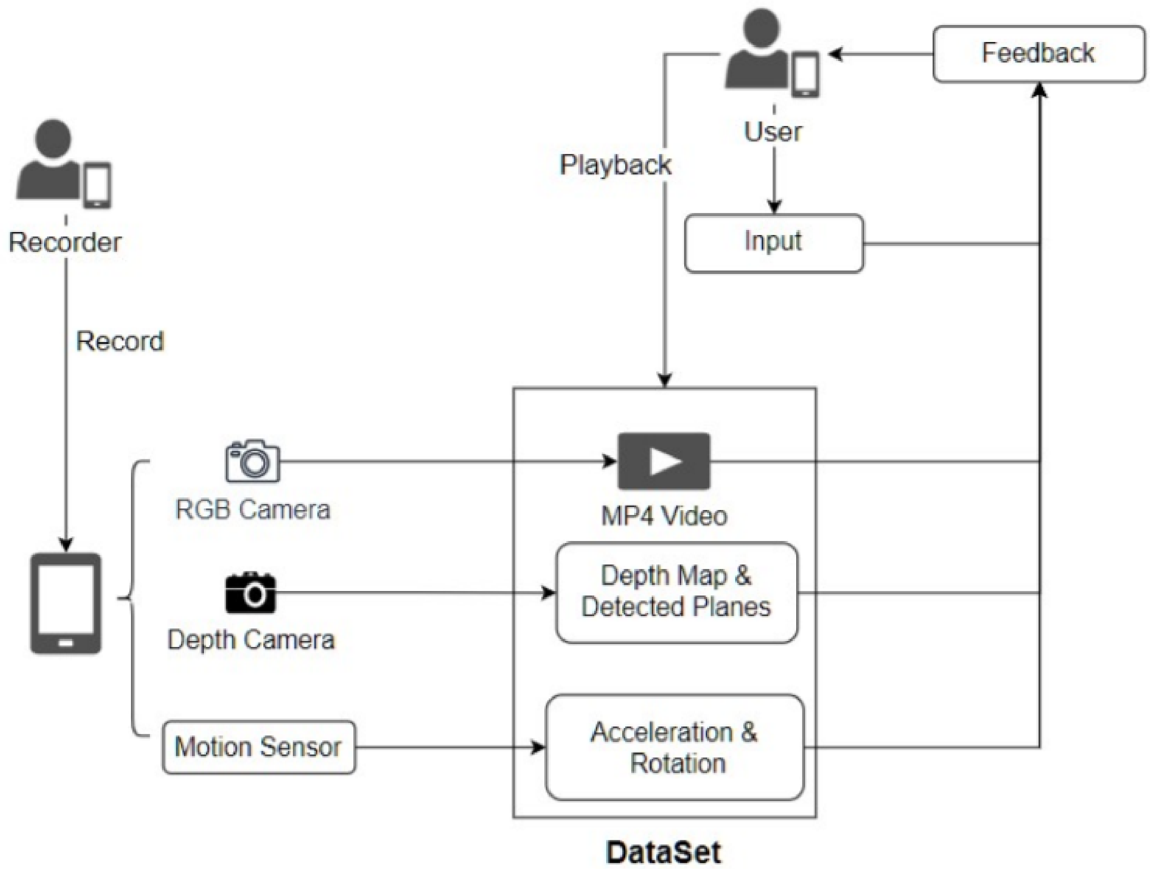


Fig. 4.3 Mechanism of Video-Based AR

Fig.4.4 shows the UI of recording and playback function. Under the buttons, there is text component to show the logs of session state.

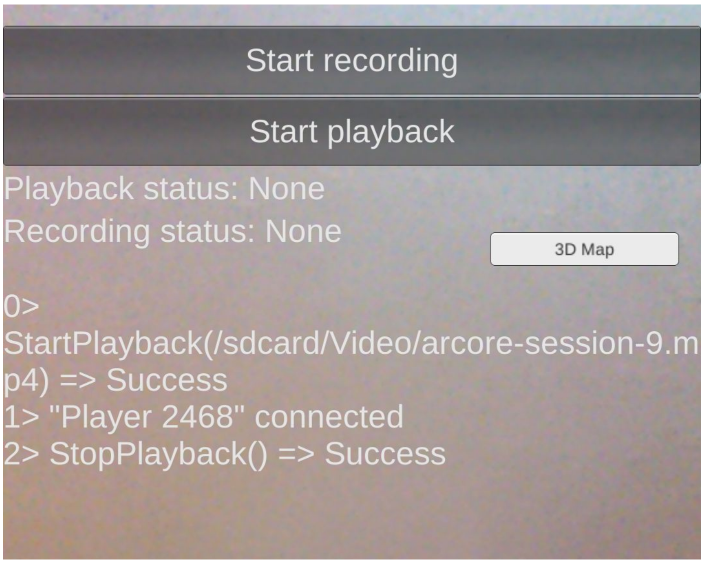


Fig. 4.4 UI of Recording and Playback

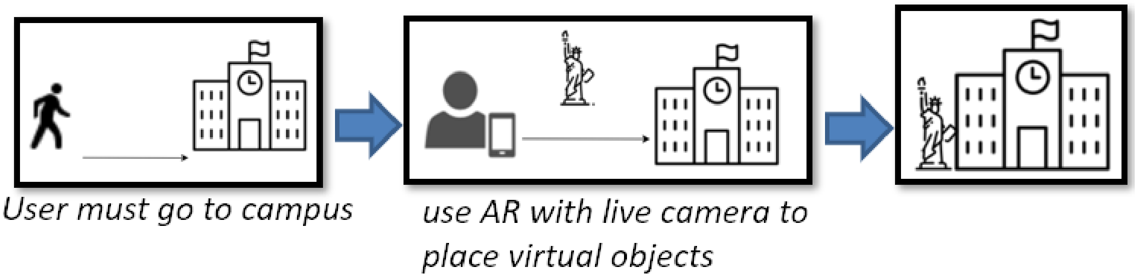
Recording

A general idea of remote AR is to transmit all the above data to another remote device through the network in real time to achieve real-time remote AR. While on the base of it, the video-based AR approach is to package all the above data integrated with time information. This function will be only opened for the system manager who will create the video for visitors.

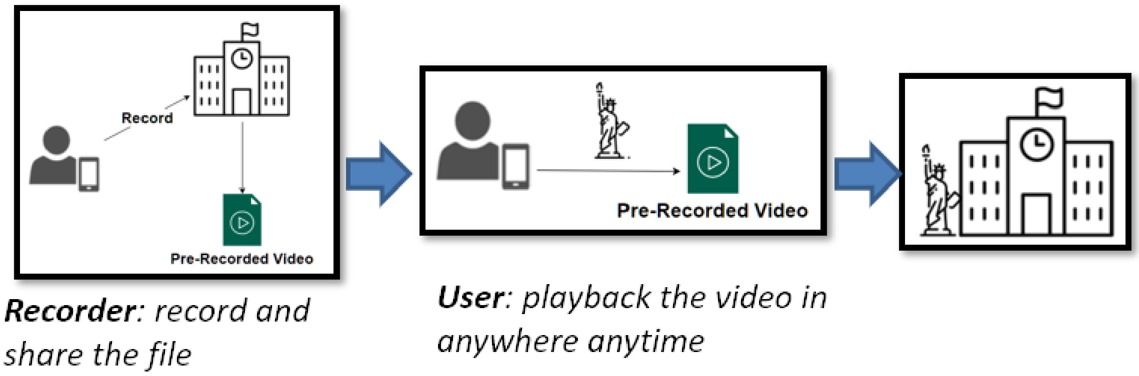
Playback

As the video player restores the real image of the video according to the time axis, the AR data corresponding to the current frame can be restored by using the packed depth information and pose information to restore the AR session corresponding to the current frame while playing the data set. Even though, we set a button of playback, but usually for the normal visitors, they will automatically start the playback from the same time when the all group members are ready.

Fig.4.3 shows the mechanism of video-based AR. The system will package the intermediate data into a data set. This data set is encoded in the format of MP4 video file. It can be shared and used by other devices. The system has two actors, the recorded and the user. We assume the recorder is the system manager who will create the source video by the "Record" action and upload them to the server. And we assume the user is the target user of online open campus, that means the visitors. They can use their own phone to fetch the source video from the server and restore the information by playback the MP4 file with video-based AR capability.



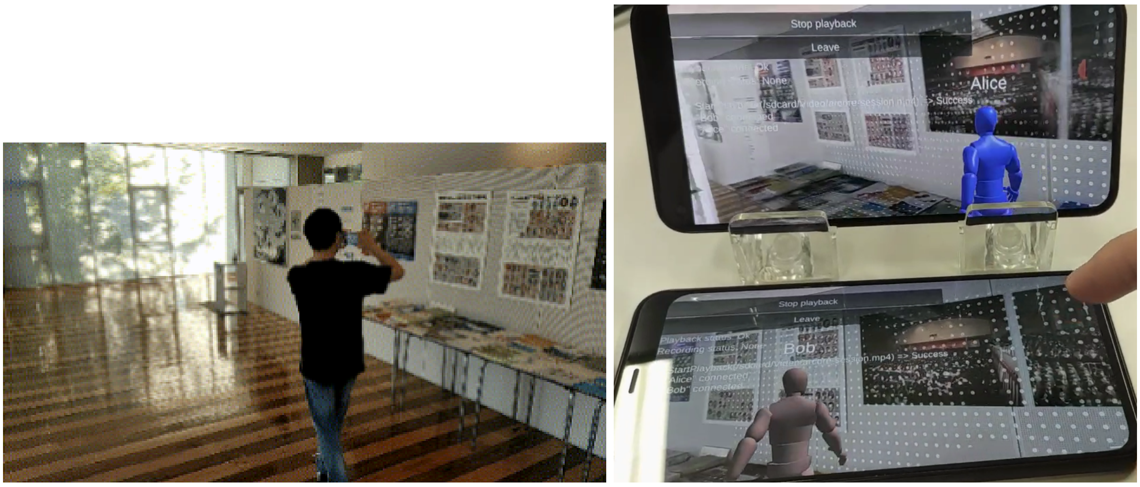
(a) General AR



(b) Video-Based AR

Fig. 4.5 Place virtual objects in campus

We use an example in Fig.4.5 to show the difference of usage between general AR and video-based AR. In the situation of general AR in Fig.4.5a, the user must go to the campus. And then use AR with a live camera to place virtual objects. However using Video-Based AR in Fig.4.5b and Fig.4.6a, we just need someone to record a video for the campus with record function and share the file with other users. For the users, they can play the pre-recorded video in a remote place at any time. As Fig.4.6b shows, when remote users playback the video, system render the avatars and the plane visualization using AR.

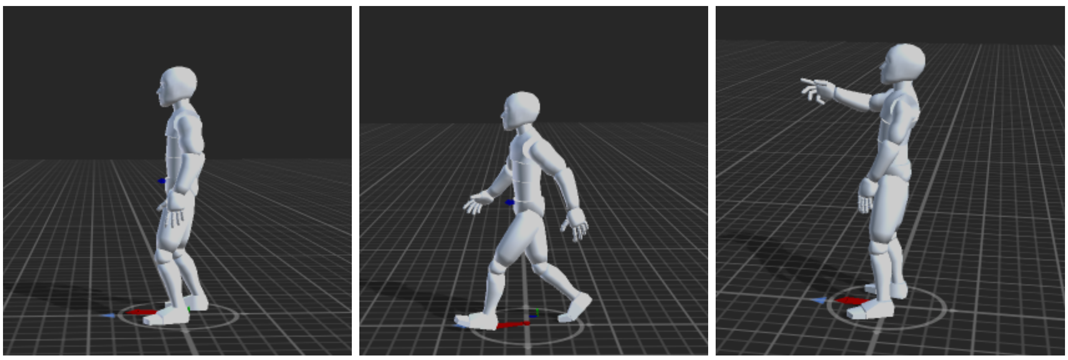


(a) Record (b) Playback

Fig. 4.6 Video-Based AR for Online Open Campus

Avatar

By placing virtual objects in the video of campus, we can involve avatars to represent users. Avatar can not only indicate the user's location, but also can play animation to communicate body language. System support the motions such as idle, walking and pointing as Fig.4.7 shows. And for difference users, we use difference color materiel on the avatar.



(a) Idle (b) Walking (c) Pointing

Fig. 4.7 Avatar Motion

Communication

Fig.4.8 shows the data transmission in real-time communication. We do not synchronize any visual data, 3D reconstruction of the scene, or motion data through the network. Each device reconstructs the AR Session using the same data set, and the data set files were shared in advance. The system only ensures that users playback the data set at the same time to keep the AR Session consistent. The data generated by user interaction is reflected in Avatar and AR annotations (AR annotations include virtual objects such as pointers, rays, and polygons, which are collectively referred to here). Based on the feature of Video-Based AR, we designed this data transmission mechanism for our remote AR that can reduce the bandwidth requirements for real-time communication. As we tested the demonstration, we found that network latency and the difference in mobile phone performance will cause the playback mode to start not completely at the same time, but still within an acceptable range.

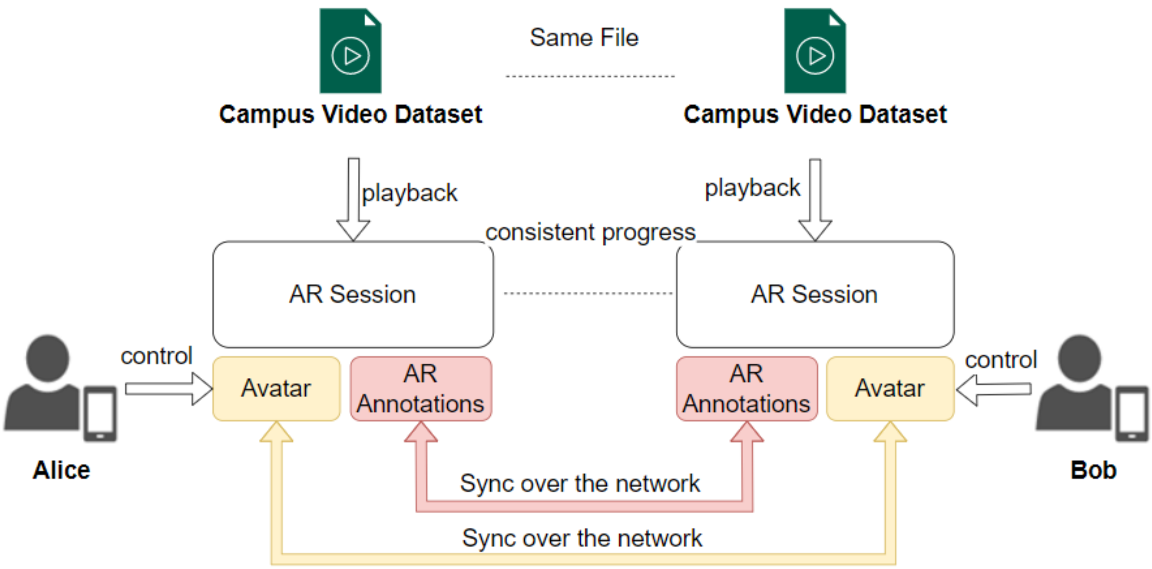


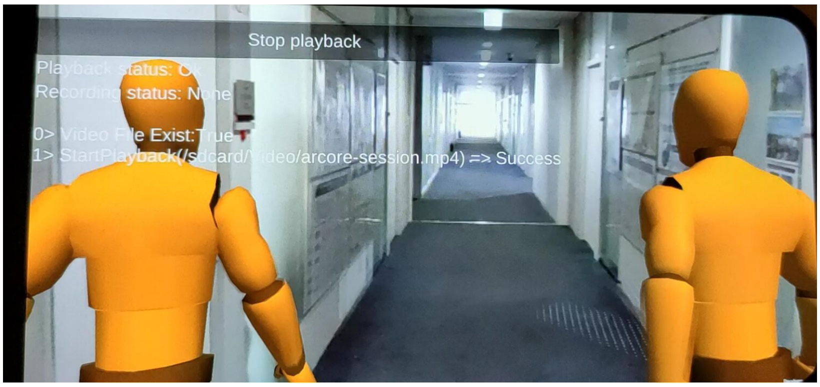
Fig. 4.8 Data Transmission in Real-time Communication

4.2.1 Group Visiting

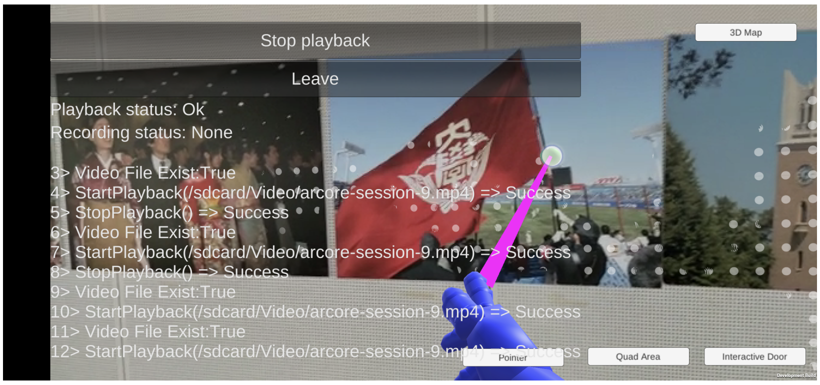
The video-based AR mode is applied to the case of group visiting, in which the places are usually public areas, such as outdoor area, student lobby and corridor.

We assume that all users start from the same place at the same time and follow together all the way, excluding the delay caused by network. In order to ensure that users start at the same time, we design a waiting room. The first person to enter the group visiting will become the room host who has a "start" button to start the group visiting when all users are ready.

In the system, users are in 1st-person view. As Fig.4.9 shows, when the user does not give any input, the avatar will keep the walking action. From the 1st-person view, there is no significant effects. Once the user point on some object by touching input, user can see the hand action from the 1st-person view. Compared to the 3rd-person view, we think that the 1st-person view can help users identify themselves when there are too many users on the screen.



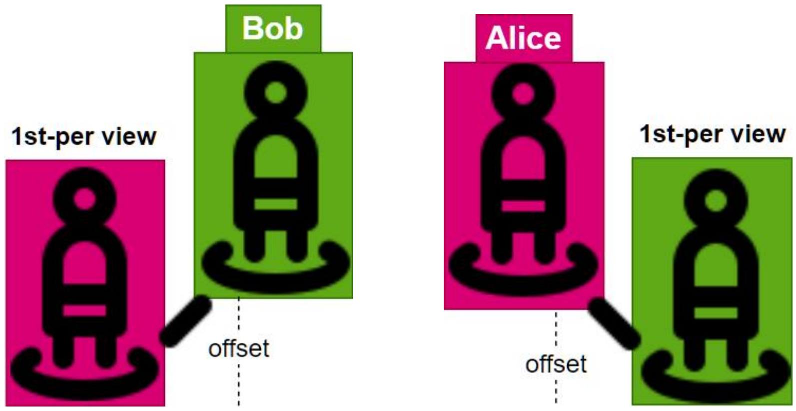
(a) Walking



(b) Pointing

Fig. 4.9 1st-Person View

Regarding the alignment of avatar in the situation of multiple users, take the current user's screen as a reference, the horizontal order is generated randomly. To optimize the display of avatars, we canceled the absolute relationship of position, but added an offset from the front direction for users other than the current user as Fig.4.10 shows.



(a) Alice's View (b) Bob's View

Fig. 4.10 Avatar Alignment

4.3 3D Model

We also use 3D model as another approach of remote AR. The difference is that 3D model will be used for room-scale areas like laboratory or meeting room. Because in video-based AR mode, user cannot control the camera of pre-recorded video, which makes it difficult to explore the facilities in more detail.

For closed room-scale areas, we scan the environment in 360 degrees by LiDAR camera to get the 3D model. The current user will be represented as an avatar from the 1st-person perspective the same as video-based AR mode, and also the avatar of others will be synchronized.

When entering the room, the system will place avatar in an initial position. The user sees the first-person perspective of this avatar from the screen. Fig.4.11 shows the first-person perspective used by the user and the third-person perspective for comparison.

In the 3D model mode, users have a higher freedom degree to control the view by themselves. So We design a virtual joystick as the interface to control the avatar's movement and direction. The user interface is shown as Fig.4.12. Similar to the video-based AR mode, users can use the avatar to walk around in the model of corresponding real space, while user can control the user perspective more freely to get a more detail observation in laboratory room.

(a) The Position of Avatar

(b) User's View

Fig. 4.11 Avatar in 3D Model

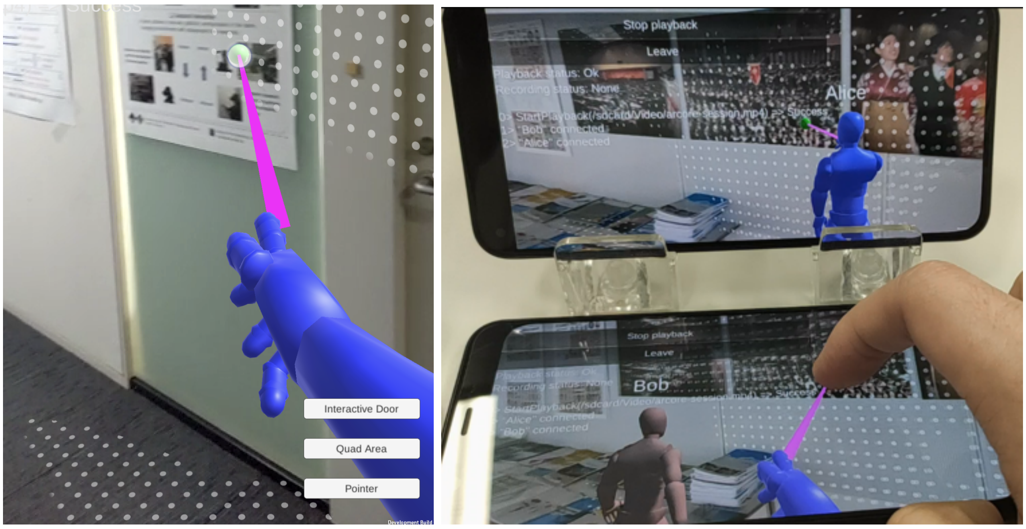
Fig. 4.12 Virtual Joystick

4.4 Sharing Functions

In order to improve real-time communication, we design the sharing functions. The function list is as follows:

- Pointing: Ray casting and avatar's hand;
- Voice Sharing: Distance adaptive volume;
- User Profile: Edit and View user basic information.

4.4.1 Pointing



(a) Point Out in 3D (b) Synchronization

Fig. 4.13 Pointing Function

Users can choose the pointer mode to use the pointing function. With AR capabilities, the system can recognize the environment in 3D and detect the plane from the video. When the user touches the touch, a virtual will be sent to hit the target position. As Fig.4.13 shows, the system renders a virtual ray and an indicator object at the hit point to help the user to point out the target. And the avatar's hand will synchronize the pointing action. The virtual ray, indicator, and actions will be synchronized to other users. So when users find some interesting place, they can use this function to share it.

Regarding the position of the indicator and the direction of the ray, the method we use is to synchronize the world position of the indicator, and then redirect the ray. Because of the network delay and the performance gap of the device, users may not be able to keep the same video playback progress. Directly synchronizing the direction of the rays may cause inconsistency in pointing to the target.

4.4.2 Voice Sharing

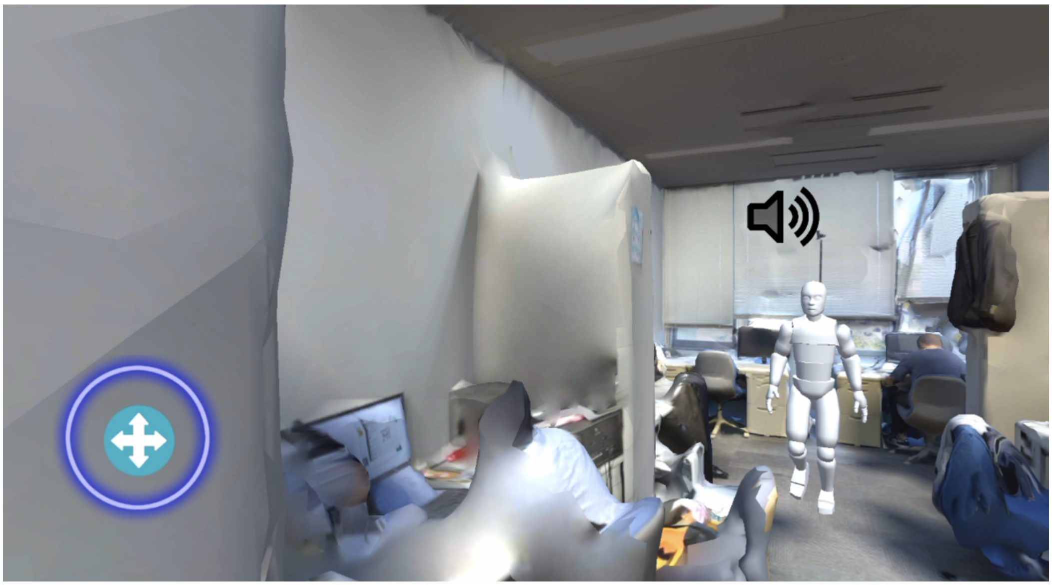


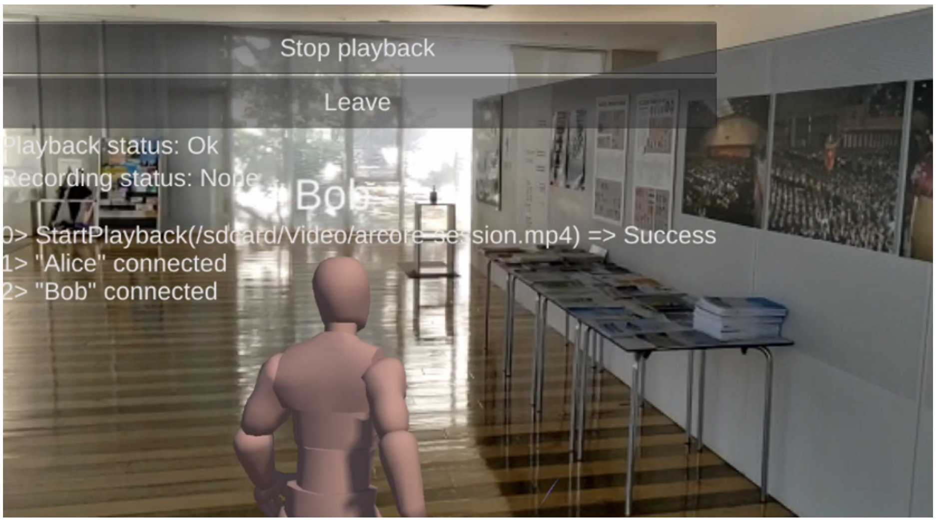
Fig. 4.14 Voice Sharing

We design the voice sharing function to allow users to use voice chat in real-time with others in the same area. In addition to the basic communication function, we use the position information of AR to make the receiver can hear the volume decreases as the distance from the sound source increases. We want to use this distance adaptive volume to simulate the scene of the real world. By controlling the distance and volume, users can create a private talk in a natural way.

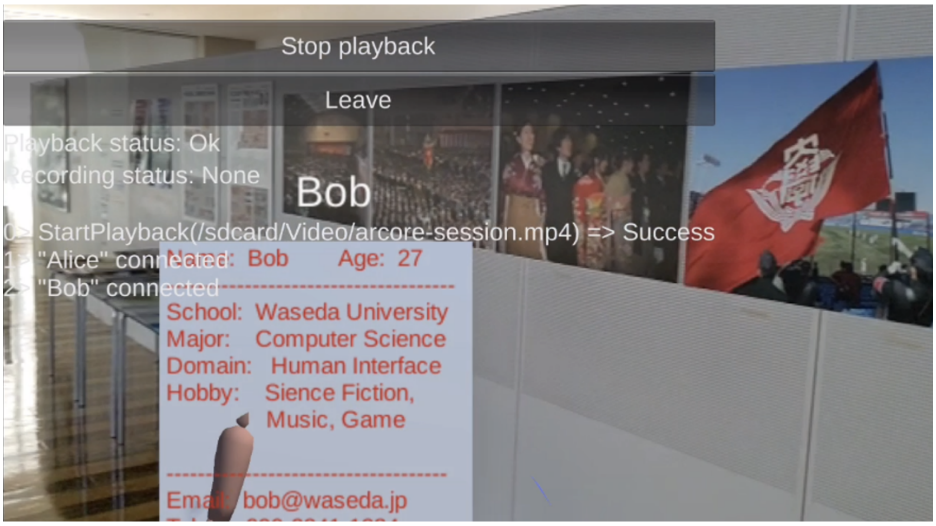
When the volume of the speaker is detected to reach a certain threshold, the system will display the volume icon on the top of the avatar. The volume icon can be seen by others as Fig.4.14 shows.

4.4.3 User Profile

When login into the system, the user will be asked to edit some basic information as the profile. During the online open campus, the profile can be viewed by touching its avatar. This function is expected to help users quickly find some common ground with others to promote their communication. Fig.4.15 shows the virtual board displayed on the avatar when touching it.



(a) Touch the Avatar



(b) Show the Profile

Fig. 4.15 User Profile

4.5 3D Map Navigation

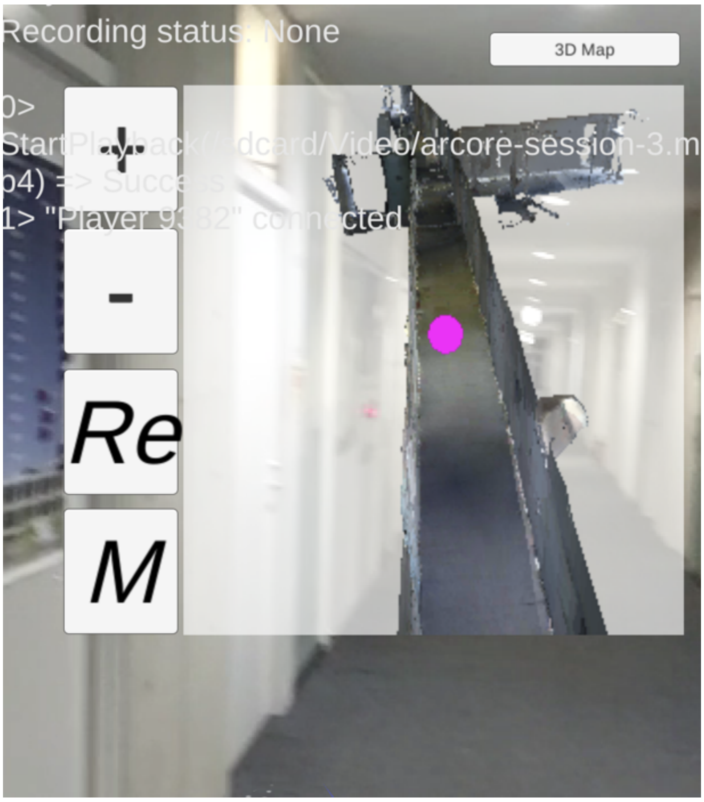


Fig. 4.16 3D Map Navigation

The navigation function can always improve the user experience by indicating the current position on a map. We use the 3D model generated by scanning the same place of video to make the corresponding 3D map.

Fig.5.16 shows our 3D map made by a 3D model. By synchronizing the position of the indicator with the dynamic current position in the video, the user can intuitively understand the current position and the route to visit in the video.

4.6 Door Interaction

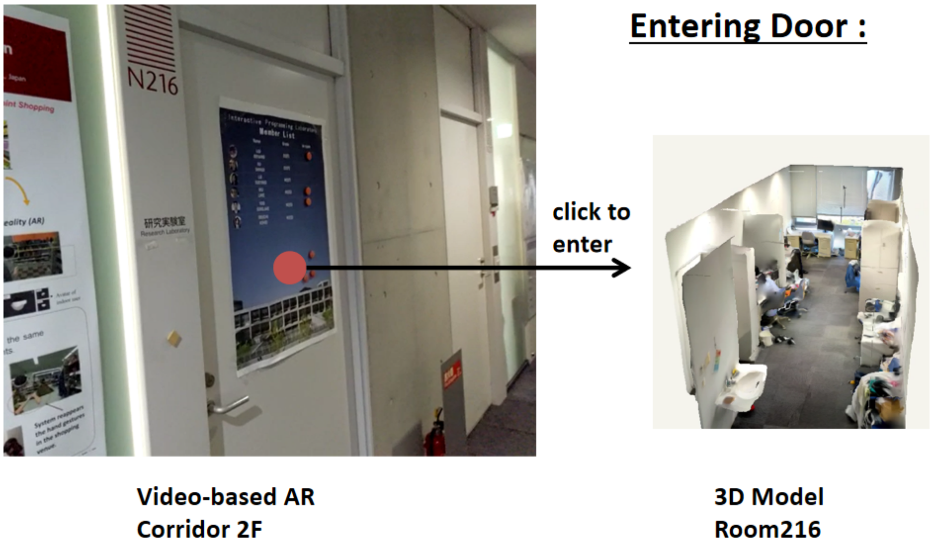
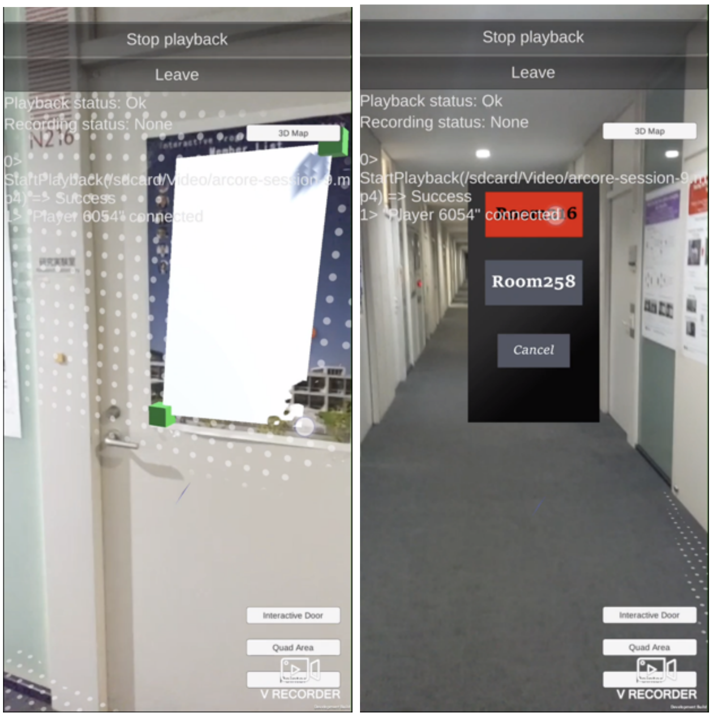


Fig. 4.17 Door Interaction

We design the door interaction to integrate the video-based AR mode with the 3D model mode. As it is shown in Fig.4.17, users can interact with doors in the video to enter another room, which means the system will switch to 3D model mode of this room instead of video. We expect this function to increase user freedom to choose their interesting laboratory room to visit.

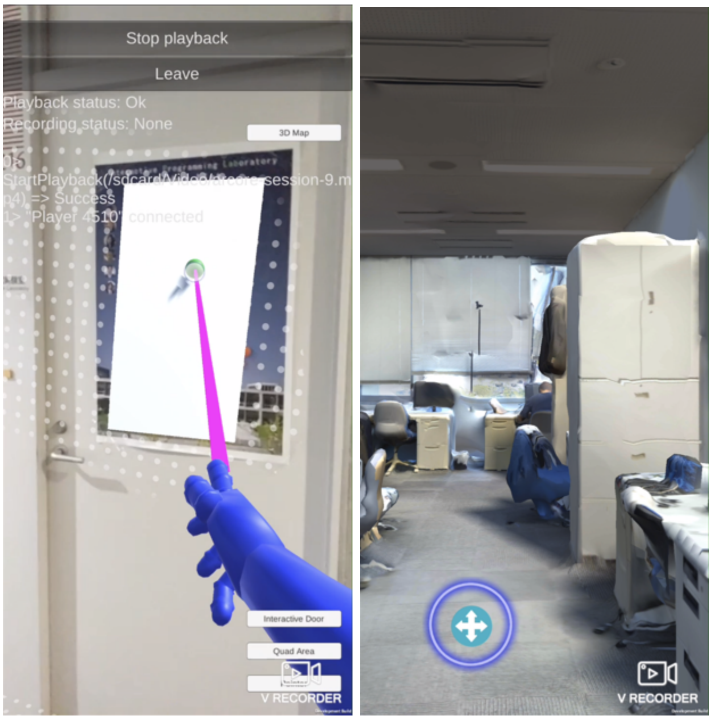
Before that, the system administrator needs to specify the position of the interactive area of the door and the target 3D model of the room after recording the video. As Fig.4.18a shows, administrator can specify the area by dragging the 4 vertices of the quad. Then in Fig.4.18b, chooses a target from the room list.

Fig.4.19 shows that the scene will be changed to the target room216 and the system mode will be switch to 3D mode with a virtual joystick when the user point on the interactive door.



(a) Specify the Area (b) Set the Target Room

Fig. 4.18 Admin: Create Area



(a) Point On (b) Enter the Target Room

Fig. 4.19 User: Interact with Door

Chapter 5

System Implementation

We implement the system on Android phones. The Google Pixel4 and Pixel5 have IMU sensors to capture device motion and support AR applications based on ARCore. About the development environment, we use Unity as the platform and use the cross-platform unified APIs provided by ARFoundation framework.

5.1 ARCore Recording and Playback

For the implementation of video-based AR, ARCore[2] provides the recording and playback APIs to achieve this function. In version 4.2.0 of ARFoundation SDK, the functions are located in the "UnityEngine.XR.ARCore.ARCoreSessionSubsystem" class. As the configuration, we add an ARSession component into the unity project as Fig.5.1 shows.

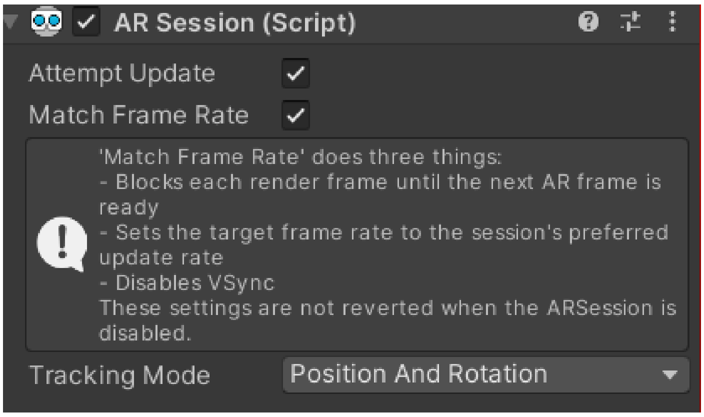


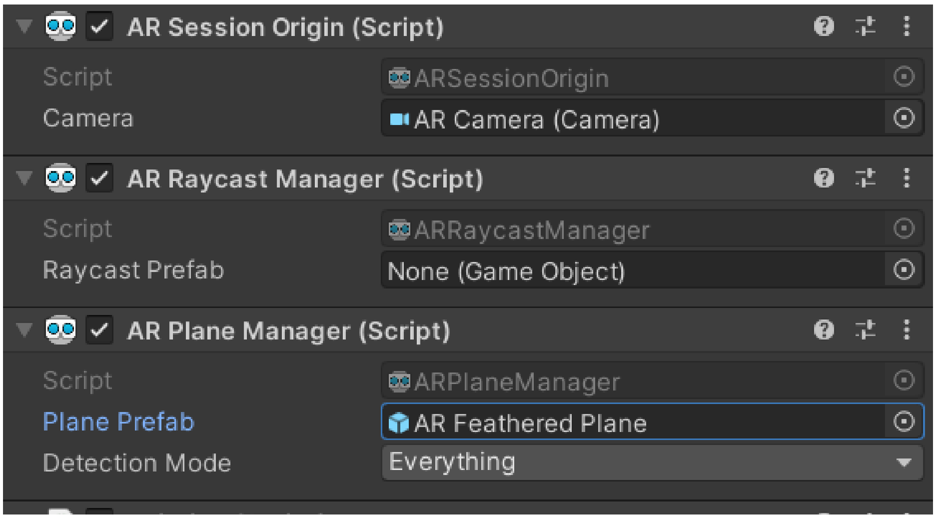
Fig. 5.1 ARSession Component

In Fig.5.2, after creating the ARsession, we can call the recording API by setting a configuration about file path and orientation or call the playback API by specifying the path of the data set. The variable named subsystem is the instance as ARCoreSessionSubsystem class that inherits from ARSession. It provides the functions to start or stop the recording and playback and return a structure containing status information.

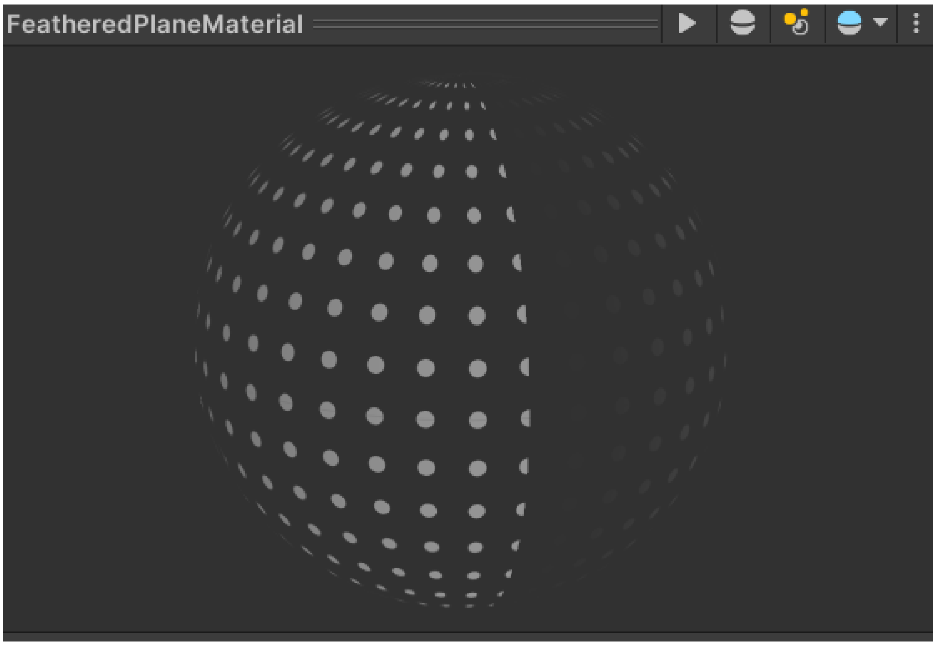
```
{
    if (playbackStatus != ArPlaybackStatus.Finished && GUILayout.Button("Start recording"))
    {//Start Recording
        using (var config = new ArRecordingConfig(session))
        {
            //Set the path of dataset file
            config.SetMp4DatasetFilePath(session, m_Mp4Path);
            //Use the same orientation of video with the screen
            config.SetRecordingRotation(session, GetRotation());
            //Use the recording API
            var status = subsystem.StartRecording(config);
            Log($"StartRecording to {config.GetMp4DatasetFilePath(session)} => {status}");
        }
    }

    if (File.Exists(m_Mp4Path) && GUILayout.Button("Start playback"))
    {//Start Playback
        Log("Video File Exist:" + File.Exists(m_Mp4Path));
        //Use the playback API
        var status = subsystem.StartPlayback(m_Mp4Path);
        Log($"StartPlayback({m_Mp4Path}) => {status}");
        //Initiate the avatar's action
        foreach(AvatarController avatar in AvatarManager.avatarList)
        {
            avatar.walk();
        }
    }
}
```

Fig. 5.2 Use Recording and PlayBack APIs

Once completed replacing the live camera AR Session with the video-based AR Session, configuring AR functionality with ARCore is the same as a general AR system. Here is the example of configuring the plane detection function. As shown in the Fig.5.3a, we add the AR Plane Manager component to the AR Session Origin. And Fig.5.3b shows the plane visualization material we use.



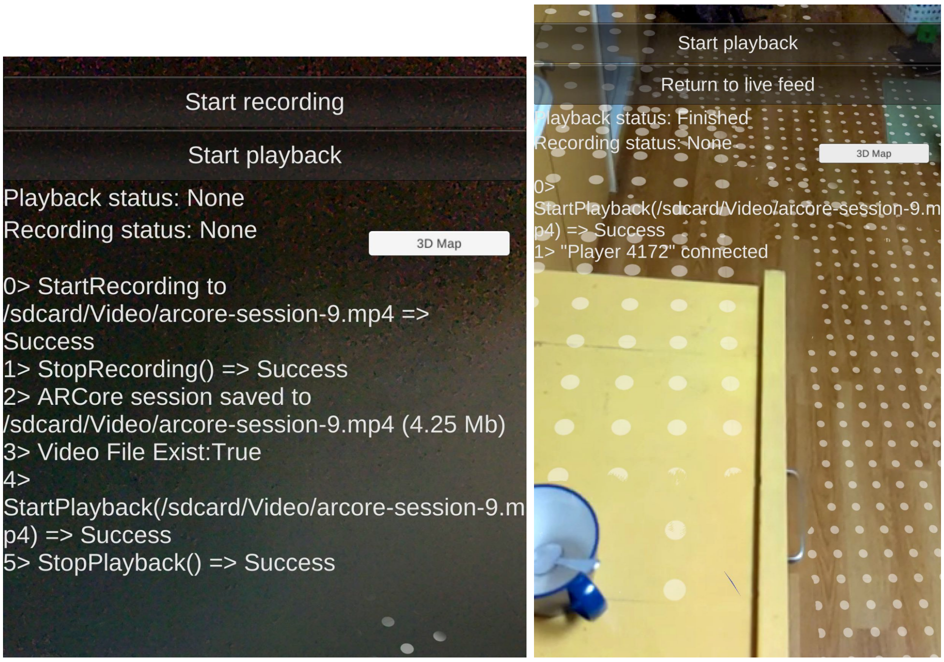
(a) AR Plane Manager Component



(b) Visualization of Detected Planes

Fig. 5.3 Configure Plane Detection

Finally, We bind the script on the GUI buttons and print out the logs of the result as shown in Fig.5.4a. When playing back a data set, the system can use the same AR capabilities as in general AR. Fig.5.4b shows the result of using the plane detection of AR in playback status.



(a) Logs and GUI (b) Plane Detection in Playback

Fig. 5.4 Use AR in the Video

5.2 3D Model Generation

LiDAR is a distance measurement method based on the reflection time difference of light, which has recently been installed in some consumer products. Therefore, many applications based on LiDAR to scan models have appeared. We use an iPad Pro and the application Polycam[28] to scan the laboratory room. As Fig.5.5 shows, the meshes of the environment are generated, and the places in blue color are to be supplemented by moving the camera. The complete room model can be generated by moving the camera to scan all the corners of the laboratory room.

Fig. 5.5 Scan the Room

We import the 3d model and instantiate the avatars in the Unity scene. After we adjusted the scale of the model and placed an avatar in at an appropriate starting point to adapt to the size of the avatar, we get result as shown in Fig.5.7. In order to make avatar can be controlled by the virtual joystick, we make the hierarchy of objects like Fig.5.6 shows. We use a script to make the main camera move with the joystick drag, and make the avatar attach to the main camera as a child object. Then keep the avatar in the room models by adding collisions.

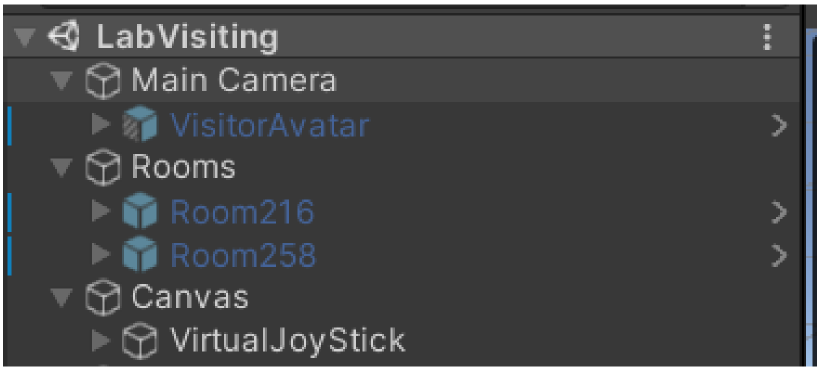
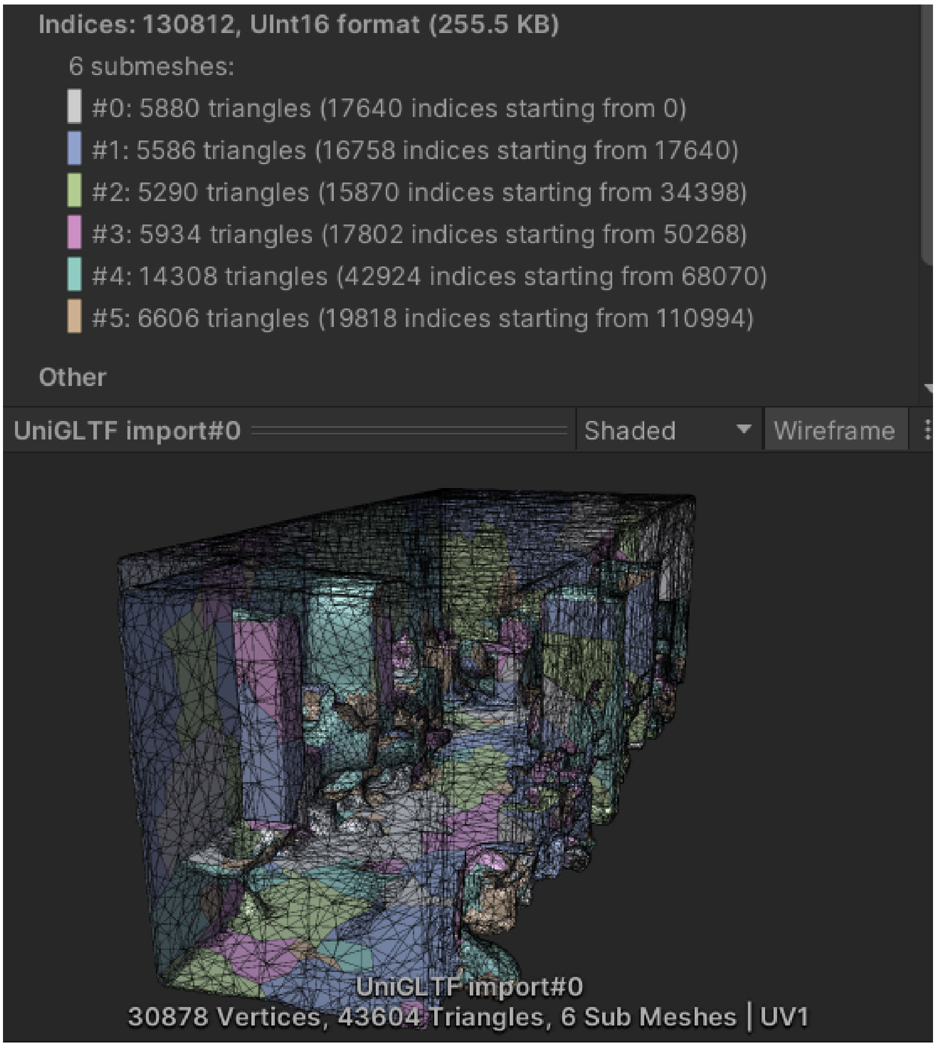


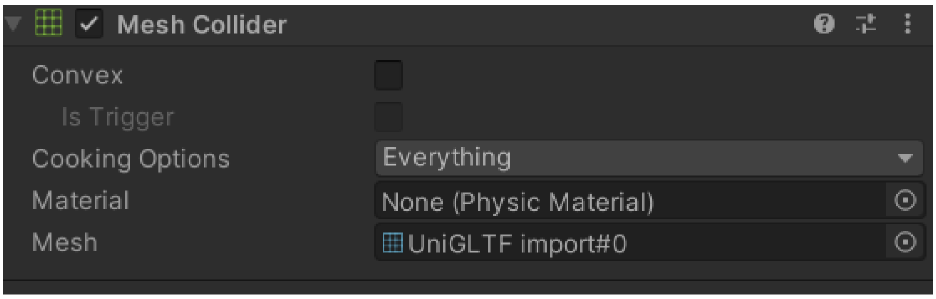
Fig. 5.6 Hierarchy of Avatar, Model and VirtualJoyStick

Fig. 5.7 Import 3D Model and Avatar into System

The format of 3D model file is GLB. After we import the file into Unity, we will get a separated mesh file of it as Fig.5.8a shows. In order to ensure that the avatar does not move through the wall, we add a Mesh Collider component to the object of room and set the mesh with that mesh file as Fig.5.8 shows.



(a) Mesh of GLB File



(b) Mesh Collider

Fig. 5.8 Use Mesh for Collisions

Because the avatar we set also has a cube-shaped box collider component, when the collider of the wall collides with the collider of the avatar, the Unity physics engine will block the avatar from continuing to move, so as to prevent the effect of passing through the wall, as shown in Fig.5.9.

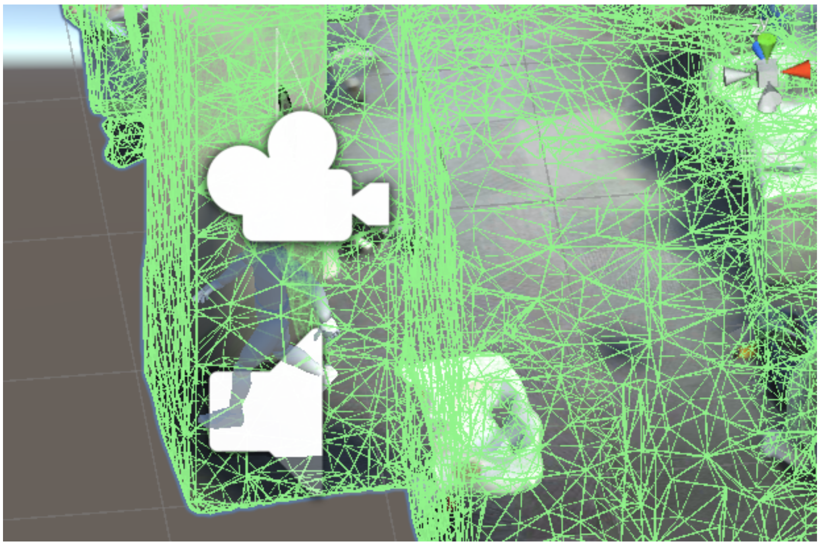


Fig. 5.9 Avatar Blocked by Mesh Collider

5.3 Pointing

When the user touches the screen, the system will render an indicator to show the hit point and direction. Fig.5.10 shows the visualization of the indicator. It consists of a ray and a sphere object. We will first calculate the position of the hit point to place the sphere indicator. Then use the position of the end of the index finger (Fig.5.11) of the current user's avatar to connect it with the hit point to render the line.

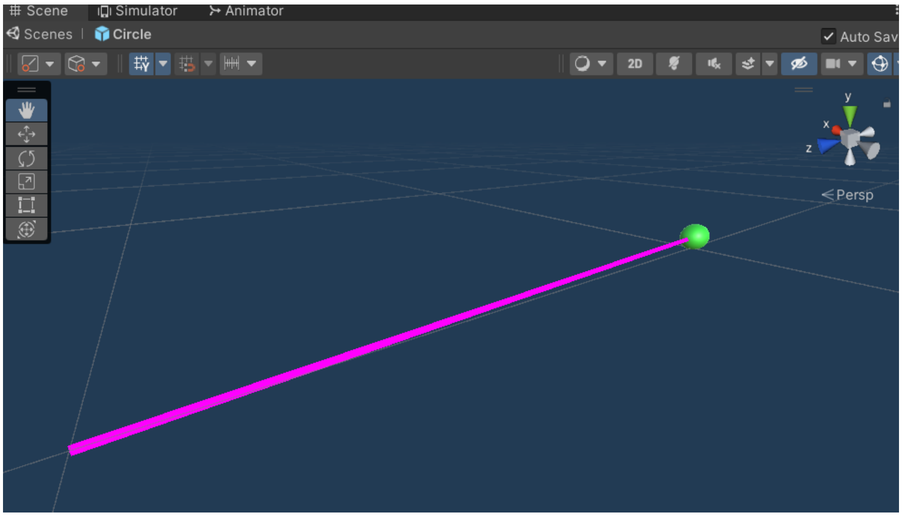


Fig. 5.10 Visualization of Indicator

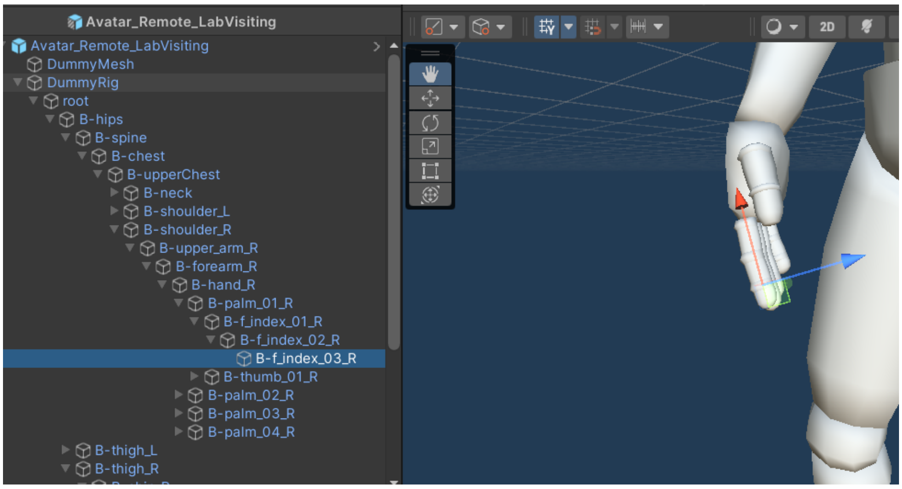


Fig. 5.11 Index Finger of Avatar

The calculation of the hit point is based on the result of the recognition of the real world. With the ability of plane detection of AR, we simulate launching an invisible virtual ray first, and then sort the detected planes by distance, and select the intersection of the ray and the nearest plane as the hit point. It can guarantee the hit point will be only placed on the surface of real objects.

5.4 Multiplayer

For the multiplayer function, we use the service of Photon PUN2. It concludes the SDK for unity client and an optional cloud server or the SDK for own private server.

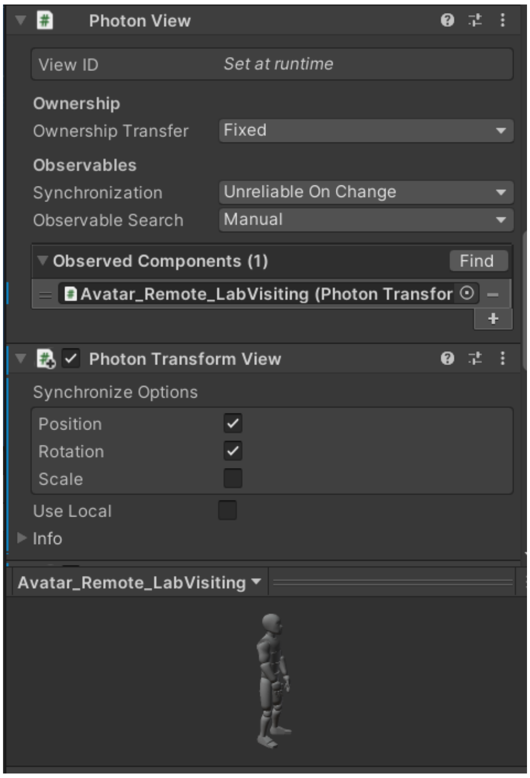


Fig. 5.12 Set Synchronization Object

Fig.5.12 shows that we set the PhotonView component on the avatar and we check the position and rotation in the Photon Transform View that is the information we expect to synchronize through the network.

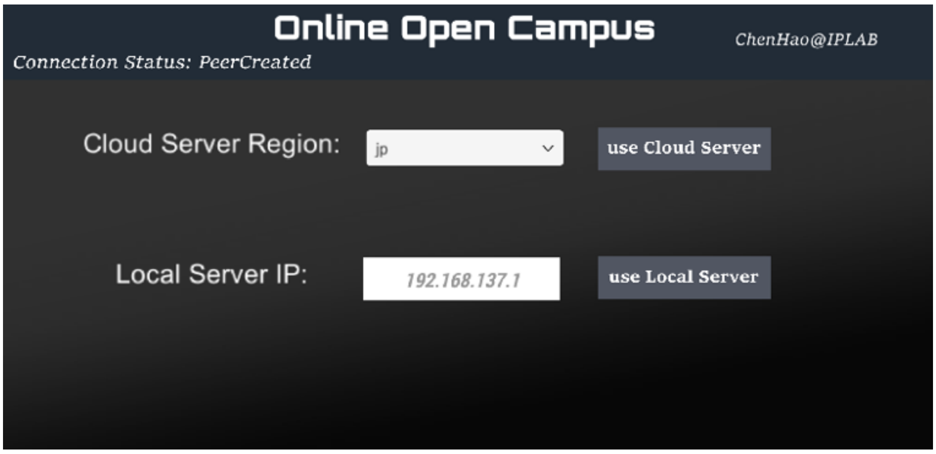


Fig. 5.13 Server Option

We comprehensively considered the flexibility and stability of the server, the system supports the use of cloud services and private servers at the same time. We created a setting GUI for switching as shown in Fig.5.13. The logs of the local server are shown in Fig.5.14.

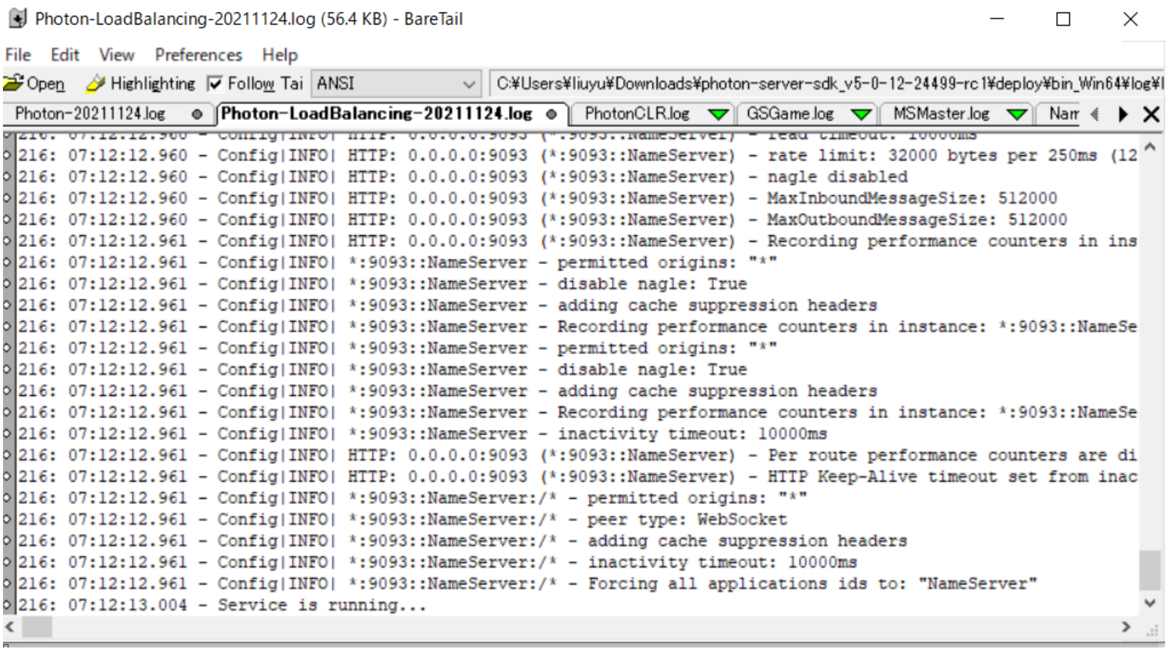
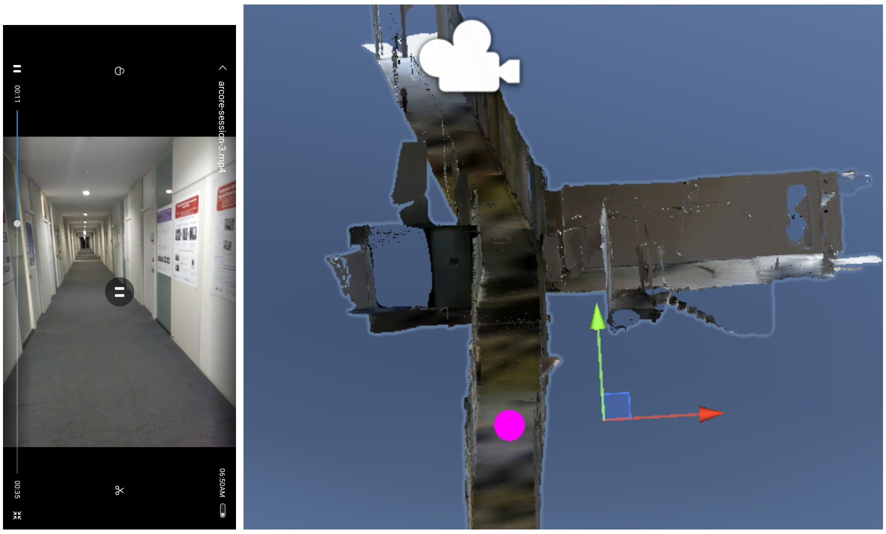


Fig. 5.14 Private Server Logs

5.5 3D Map Navigation

First, we used the same LiDAR method to get the model of the real scene. It should be noted that the scene here is not the laboratory room but the scene that appears in the video in Group Visiting, because we will use this model as the navigation map in the video playback mode. Fig.5.15 shows a video data set recorded by the system and the corresponding 3D model of the scene in the video.



(a) Video File (b) Model of Video Scene

Fig. 5.15 Video and 3D Map

Then we can set a red spherical object in the model in Fig.5.15b. It is the indicator of the current location of the user. In the programs of ARCore, the current location can be simply acquired by the main camera. The pose of the main camera is also continuously updated by the motion data from the data set in playback mode. Therefore, we can use the world pose of the main camera to update the local pose of the indicator in the map model with a parameter used to adjust the scale to achieve synchronization. Fig.5.16 shows the result of combining video-based AR and 3D map navigation.

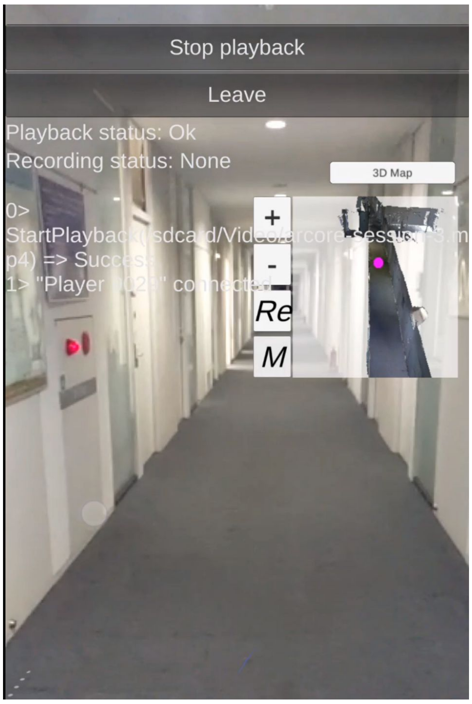


Fig. 5.16 3D Map Navigation

5.6 Door Interaction

In this section, we will introduce the implementation of the door interaction function from the steps of creating, saving, and loading the interactive area.

Creating

We assume the interactive area is quad. Fig.5.17 shows how to create the area by touching the screen. We use the touch event and its parameters to determine the different types of operation. When the user selected 3 points, the system will automatically place the remaining one point at the symmetrical position.

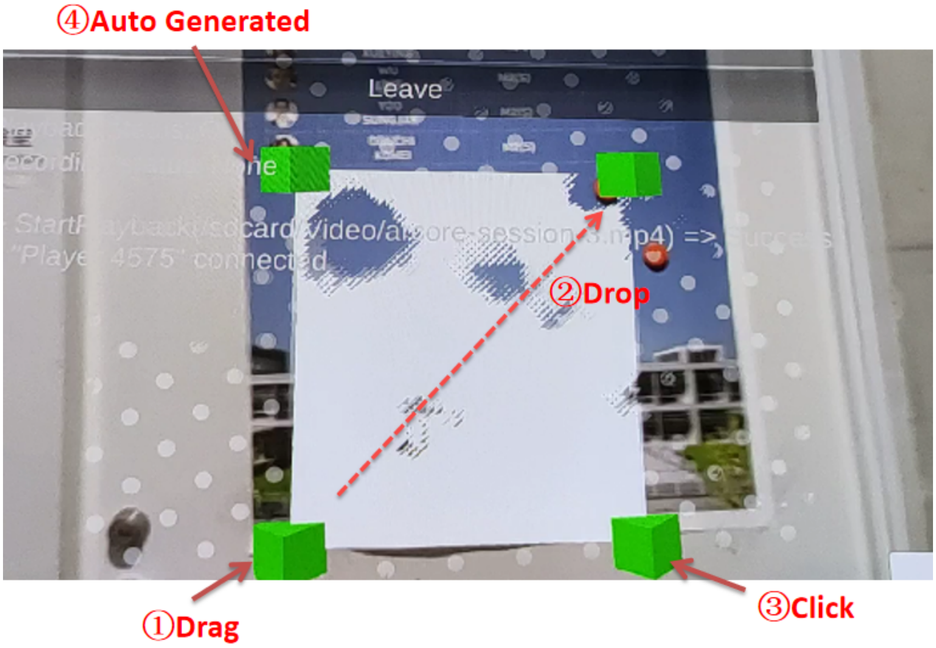


Fig. 5.17 Create an Area

Saving

The position of the door is specified by the system manager. So the data need to be persisted, we use SQ-Lite to save the data. We use the code in Fig.5.18a to create a connection to the SQ-Lite database and create the "Area" table when initial. The same as creating a table, we also use SQL statements to save and update data. Fig.5.18b shows the overview of the data table when actual use.

```
private const string init = @"CREATE TABLE [Area] (
    [Id] TEXT NOT NULL PRIMARY KEY,
   [Scene] TEXT,
   [Point1] TEXT,
    [Point2] TEXT,
    [Point3] TEXT,
    [Point4] TEXT,
   [Comment] TEXT,
   [Created] TEXT,
   [Modified] TEXT
    );";

// Start is called before the first frame update
void Awake()
{
    //Create the connection to sqlite
    using (Database = new SQLite("SQLite.db", init, DataFilePath))
    {
        Debug.Log("Connect SQLite Successful");
```

(a) Connect and initialize Database

オブジェクト　Area @main (Mobile) - テーブル

トランザクションを開始　テキスト　フィルター　ソート　インポート　エクスポート

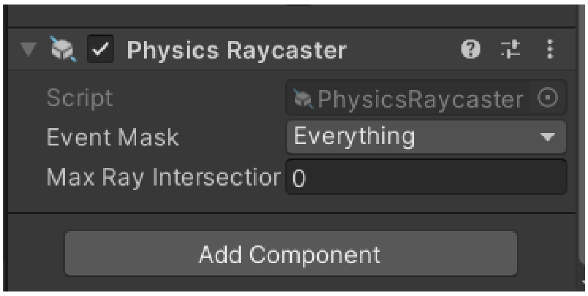
Id	Scene	Point1	Point2	Point3	Point4	Comment	Created
a32476b0-2b06-4419-	Room216	(1.21, -0.36, 3.22)	(1.25, -0.29, 2.78)	(1.22, 0.29, 3.10)	(1.27, 0.35, 2.65)	Some Comment	(Null)
1059dd20-e572-4fac-	Room258	(-1.86, -0.27, 11.42)	(-1.39, -0.26, 8.32)	(-1.35, 0.36, 7.80)	(-1.39, 0.39, 8.31)	Some Comment	(Null)

(b) Area Table

Fig. 5.18 Data Persistence

Loading

When loading the scene of video-based AR, the system will not only render the interactive areas from the database but also will call a function to enable the interaction of the door. As Fig. 5.19a shows, we set the "Physics Raycaster" component on the main AR camera to utilize the ray-casting function of unity engine. And in Fig.5.19b, we create an event trigger at the corresponding position of the real door to receive the user's click event, thus realizing the interactive function of the door.



(a) Add Raycaster Component

```
/// <summary>
/// Add Interaction by PointerDown event
/// </summary>
public void AddInteraction()
{
    EventTrigger trigger = this.quad.AddComponent<EventTrigger>();
    EventTrigger.Entry entry = new EventTrigger.Entry();
    entry.eventID = EventTriggerType.PointerDown;
    entry.callback.AddListener((data) => { OnPointerDownDelegate((PointerEventData)data); });
    trigger.triggers.Add(entry);

}
```

(b) Add Event Trigger

Fig. 5.19 Enable Interaction on Door

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In conclusion, in this research, we discussed the problem of insufficient communication in the existing online open campus system, compared the existing video stream-based methods and 3D model-based methods, and proposed novel ideas for using remote AR. And compared a variety of remote AR methods, we adopted a specific method of fusing Video-based AR and 3D models. That is to improve the user experience, users can visit the remote campus at any place and time, and can intuitively interact with real objects. At the same time, only a small amount of bandwidth is required because only the interactive data is synchronized.

Considering the requirement of multi-user communication on the online open campus, we use avatar to represent users. The avatar supports some motion to communicate. Based on avatar, we also designed a sharing function for users to improve communication. Users can share position of interest not only through sounds and actions, but also through sharing pointers on the surface of real objects.

6.2 Future Work

We think there is still room for improvement in user experience. For example, the FoV of the mobile phone is too small and the visitor watching the video cannot move the angle of view in video playback. Some other remote AR systems use the combination of 360-degree video to increase the FoV, but there are still challenges in using ARcore to integrate 360-degree video. Furthermore, it is also very interesting to allow remote users to interact with real-time users in the field. However, the videos watched by remote users and the real-time users in the site are at different times. We expect that through the motion capture technologies, the on-site users can appear in the form of avatars. In addition, the user study of this thesis is preliminary, so we will upgrade it in the future.

References

[1] Zoom, Zoom meeting, Internet WWW-page, URL: https://zoom.us/, 2021.

[2] Google, Introduction to recording and playback on ar foundation for android, Internet WWW-page, URL: https://developers.google.com/ar/develop/unity-arf/recording-and-playback/introduction, 2021.

[3] B. Nuernberger, E. Ofek, H. Benko, A. D. Wilson, Snaptoreality: Aligning augmented reality to the real world, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, Association for Computing Machinery, 2016, p. 1233–1244. doi:10.1145/2858036.2858250.

[4] D. Chekhlov, A. P. Gee, A. Calway, W. Mayol-Cuevas, Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam, in: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007, pp. 153–156. doi:10.1109/ISMAR.2007.4538840.

[5] M. Joachimczak, J. Liu, H. Ando, Real-time mixed-reality telepresence via 3d reconstruction with hololens and commodity depth sensors, in: Proceedings of the 19th ACM International Conference on Multimodal Interaction, ICMI '17, Association for Computing Machinery, 2017, p. 514–515. doi:10.1145/3136755.3143031.

[6] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, S. Izadi, Holoportation: Virtual 3d teleportation in real-time, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16, Association for Computing Machinery, 2016, p. 741–754. doi:10.1145/2984511.2984517.

[7] T. Teo, L. Lawrence, G. A. Lee, M. Billinghurst, M. Adcock, Mixed Reality Remote Collaboration Combining 360 Video and 3D Reconstruction, Association for Computing Machinery, 2019, p. 1–14. doi:10.1145/3290605.3300431.

[8] L. Gao, H. Bai, G. Lee, M. Billinghurst, An oriented point-cloud view for mr remote collaboration, in: SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications, SA '16, Association for Computing Machinery, 2016. doi:10.1145/2999508.2999531.

[9] F. Tecchia, L. Alem, W. Huang, 3d helping hands: A gesture based mr system for remote collaboration, in: Proceedings of the 11th ACM SIGGRAPH International Conference

on Virtual-Reality Continuum and Its Applications in Industry, VRCAI '12, Association for Computing Machinery, 2012, p. 323–328. doi:10.1145/2407516.2407590.

[10] Z. Wang, C. Nguyen, P. Asente, J. Dorsey, DistanciAR: Authoring Site-Specific Augmented Reality Experiences for Remote Environments, CHI '21, Association for Computing Machinery, 2021. doi:10.1145/3411764.3445552.

[11] T. Piumsomboon, G. A. Lee, J. D. Hart, B. Ens, R. W. Lindeman, B. H. Thomas, M. Billinghurst, Mini-Me: An Adaptive Avatar for Mixed Reality Remote Collaboration, Association for Computing Machinery, 2018, p. 1–13. doi:10.1145/3173574.3173620.

[12] L. Lab, Second life, Internet WWW-page, URL: https://secondlife.com, 2003.

[13] M. Adcock, S. Anderson, B. Thomas, Remotefusion: Real time depth camera fusion for remote collaboration on physical tasks, VRCAI '13, Association for Computing Machinery, 2013, p. 235–242. doi:10.1145/2534329.2534331.

[14] S. Utzig, R. Kaps, S. M. Azeem, A. Gerndt, Augmented reality for remote collaboration in aircraft maintenance tasks, in: 2019 IEEE Aerospace Conference, 2019, pp. 1–10. doi:10.1109/AERO.2019.8742228.

[15] A. Nassani, L. Zhang, H. Bai, M. Billinghurst, ShowMeAround: Giving Virtual Tours Using Live 360 Video, Association for Computing Machinery, 2021. doi:10.1145/3411763.3451555.

[16] S. Kim, G. Lee, W. Huang, H. Kim, W. Woo, M. Billinghurst, Evaluating the Combination of Visual Communication Cues for HMD-Based Mixed Reality Remote Collaboration, Association for Computing Machinery, 2019, p. 1–13. doi:10.1145/3290605.3300403.

[17] T. Teo, G. A. Lee, M. Billinghurst, M. Adcock, Hand gestures and visual annotation in live 360 panorama-based mixed reality remote collaboration, in: Proceedings of the 30th Australian Conference on Computer-Human Interaction, OzCHI '18, Association for Computing Machinery, 2018, p. 406–410. doi:10.1145/3292147.3292200.

[18] G. A. Lee, T. Teo, S. Kim, M. Billinghurst, Mixed reality collaboration through sharing a live panorama, in: SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications, SA '17, Association for Computing Machinery, 2017. doi:10.1145/3132787.3139203.

[19] S. Kim, G. Lee, N. Sakata, M. Billinghurst, Improving co-presence with augmented visual communication cues for sharing experience through video conference, in: 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2014, pp. 83–92. doi:10.1109/ISMAR.2014.6948412.

[20] K. Higuch, R. Yonetani, Y. Sato, Can eye help you? effects of visualizing eye fixations on remote collaboration scenarios for physical tasks, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, Association for Computing Machinery, 2016, p. 5180–5190. doi:10.1145/2858036.2858438.

[21] L. Boffi, G. Mincolelli, S. Bertucci, F. Pes, M. Garofoli, L. Gammarota, Co-Drive: Experiencing Social Virtual Travel on a Car Trip, Association for Computing Machinery, 2021. doi:10.1145/3411763.3451557.

[22] Z. Li, D. Xu, Y. Zhang, Real walking on a virtual campus: A vr-based multimedia visualization and interaction system, in: Proceedings of the 3rd International Conference on Cryptography, Security and Privacy, ICCSP '19, Association for Computing Machinery, 2019, p. 261–266. doi:10.1145/3309074.3309112.

[23] F. R. Priolo, R. J. B. Mediavillo, A. N. D. Austria, A. G. S. D. Angeles, M. C. G. Fernando, R. S. Cheng, Virtual heritage tour: A 3d interactive virtual tour musealisation application, in: Proceedings of the 3rd International Conference on Communication and Information Processing, ICCIP '17, Association for Computing Machinery, 2017, p. 190–195. doi:10.1145/3162957.3162987.

[24] S. Tsujinaga, N. Yamaguchi, J. Liu, T. Tateyama, Y. Iwamoto, Y.-W. Chen, Interactive virtual campus tour system using skeleton information from kinect, in: 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), 2018, pp. 47–50. doi:10.1109/GCCE.2018.8574856.

[25] B. Nuernberger, M. Turk, T. Höllerer, Evaluating snapping-to-photos virtual travel interfaces for 3d reconstructed visual reality, in: Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, VRST '17, Association for Computing Machinery, 2017. doi:10.1145/3139131.3139138.

[26] C. Li, Z. Cao, Lidar-stereo: Dense depth estimation from sparse lidar and stereo images, in: Proceedings of the 2020 5th International Conference on Multimedia Systems and Signal Processing, ICMSSP 2020, Association for Computing Machinery, 2020, p. 11–15. doi:10.1145/3404716.3404721.

[27] A. Sankar, S. M. Seitz, Interactive room capture on 3d-aware mobile devices, in: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST '17, Association for Computing Machinery, 2017, p. 415–426. doi:10.1145/3126594.3126629.

[28] Polycam, Polycam - lidar 3d scanner, Internet WWW-page, URL: https://poly.cam, 2021.