ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol.12 (2022) No.1, pp.129-142
DOI:10.33168/JSMS.2022.0110

Mobile AR Contents Production Technique for Long Distance Collaboration

Gongkyu Byeon, Sunjin Yu

Department of Culture and Technology Convergence, Changwon National University, Korea.

sjyu@changwon.ac.kr

Abstract. This study proposes that mobile AR users with different physical spaces can use one collaboration coordinate by synchronizing each other's coordinates. When each user uses mobile AR remote contents from a distance, it is difficult to maintain the same working environment depending on their physical environment. In order for physical spaces to maintain the same working environment in different situations, the coordinates of the virtual object must be matched to the direction of each client. To solve this problem, this paper proposes a coordinate synchronization technology using a shared coordinate flag. Through this technique, the interaction of all subsequent objects can be performed based on the shared coordinate system of the local client. The user installs a coordinate synchronization flag on a map formed based on each physical space. Through this flag, the location of the virtual object may be synchronized based on each client to perform collaboration. To verify the proposed technology, the mobile AR remote technology was implemented as a painting contents application. Three users synchronized the coordinate system together on three Android mobile devices to verify the feasibility of the technology proposed in this paper.

Keywords: Augmented Reality, Computer Vision, Mobile Phone, Remote Collaboration, Telepresence, 3D Interaction.

1. Introduction

The demand for non-face-to-face related technologies has increased due to COVID-19. Accordingly, immersive contents technology, led by AR(Augmented Reality) and VR(Virtual Reality) technology, is being used evenly in all industries, including education, medical care, shopping, manufacturing, marketing, tourism, and architecture (Kwok et al., 2021; Myung-Ha Kim et al., 2020). However, in line with the explosive hardware development, the level of technology applied to contents does not keep up, causing problems between users and developers (Suh et al., 2018). The application technology required for contents development solves problems arising from the connection between users and developers. Therefore, it can be said that it is an important task to develop application technology so that it can be used more widely for the growth of contents. VR is a technology that is similar to reality, but is produced virtually, not real. AR is a technology that combines virtual objects with real environments by overlaying them. VR technology is being developed in various ways after the initial maturity stage, but AR technology is in the growth stage. Therefore, it is necessary to gradually diversify the application technology according to the growth rate of AR technology (Pena-Rios et al., 2018). Positively, the size and outlook of the global immersive contents market by 2022 are expected to continue to expand contents development and investment in AR technology application services (Ramtohul et al., 2022). Meanwhile, as the advancement of network technology enables real-time interference with virtual objects, attempts to construct a remote collaboration system that can conduct meetings or collaborations in a immersive contents environment continue (Dupont et al., 2018). Early AR technology remote systems were implemented focusing on Face-to-Face collaboration system contents that changes objects that users look at each other in the same space. However, as the demand for non-face-to-face related technologies increased, the demand for various remote applications increased (Ifrim et al., 2020). Therefore, there is a need for a technology capable of experiencing contents to which long-distance AR technology is applied beyond short-range AR technology. AR technology has the advantage of being able to use physical features of the real world, unlike VR technology in complete virtual space. This paper proposes a shared coordinate system technology using coordinate synchronization flags to produce long-distance AR contents using the corresponding characteristics. Mobile painting applications are implemented and tested to verify the proposed method.

2. Background

In this chapter, mobile AR technology is divided into sensor-based technology and image tracking-based technology, and describes the characteristics of development frameworks for implementing AR applications. It also introduces mobile AR near-field remote technology that shares the same feature points with cloud servers by applying cloud technology as a case study.

2.1. Classification of Technical Types of Mobile AR Contents

Mobile AR contents can be classified into sensor-based technology and image tracking-based technology according to technology (Kolivand et al., 2018). Sensor-based technology does not use camera image data. It is a technology that applies location coordinate information including longitude, latitude, and altitude of a user using GPS (Global Positioning System) and IMU(Inertial Measurement Unit) sensors mounted on a mobile. GPS technology can periodically read the user's location provision information and interact on the network when obtaining data storage rights with the Internet. The general error range of GPS is about 5-10m, and the accuracy of location information is not high, so it is suitable for application to contents used in outdoor spaces larger than indoor spaces. These sensors were mainly used in location information search and directions applications. It was also used in tool functions that allow vertically and horizontally of objects or in game applications that allow augmented characters to face gravity. Examples of application of this technology include Niantic's Pokémon GO, which was released in 2016, and various AR applications with sensor-based technology have since been fabricated (Pyaeet et al., 2016). Image tracking-based technology is classified into Marker-based detection technology and Markerless-based technology. Marker-based technology is a technology that recognizes marker images with predetermined regular patterns such as QR codes through image processing technology. Since technology implementation is simpler than other technologies, it is used as a technology to produce universal AR contents, but there is a disadvantage that it cannot be used for objects other than patterns. Markerless-based technology is also called based on natural feature points, and since it analyzes and recognizes the feature points of the real environment, augmented reality can be produced only with surrounding spatial information without markers. Object recognition recognizes the shape of the object, and since there is no limit to the range of the object to be recognized, contents capable of recognizing an actual real object may be produced. Currently, research on VIO (Visual-Inertial Odometry) and SLAM (Simulane Localization And Mapping) technologies that track the location of mobile devices in the real world and recognize real objects is continuously being developed.

2.2. Mobile AR Contents Development Framework Classification

Vuforia, ARKit, and ARCore, which are development frameworks frequently used to produce AR contents, are described (Linowes et al., 2017).

PTC's Vuforia can use the VuMark tracking function and image tracking function using Natural Feature that can minimize the sense of difference from the surrounding environment, which has been a disadvantage of existing marker-based technology. In addition, there is a ground plane tracking function that allows virtual contents to be placed on a horizontal surface of the surrounding environment such as a floor or desk. VISLAM (Visual-Inertial Simulative Localization And Mapping) of Vuforia Fusion, which is shown in the center of Fig. 1, is an algorithm implemented by Vuforia that combines the advantages of VIO and SLAM. VISLAM works better in environments with less characteristics than SLAM-based tracking, and VIO solutions provide recoverable robustness when tracking is completely lost. Through this, faster tracking may be performed when estimating the distance between the camera and the ground plane.

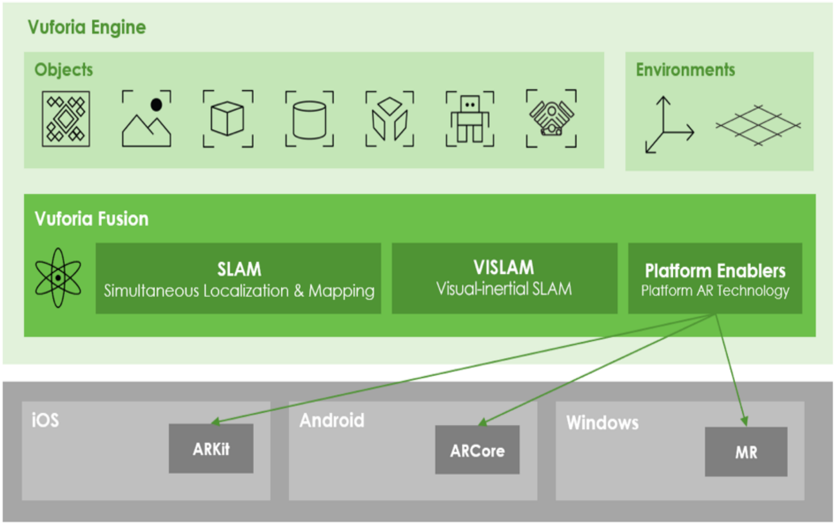

Fig. 1: Vuforia Component Diagram

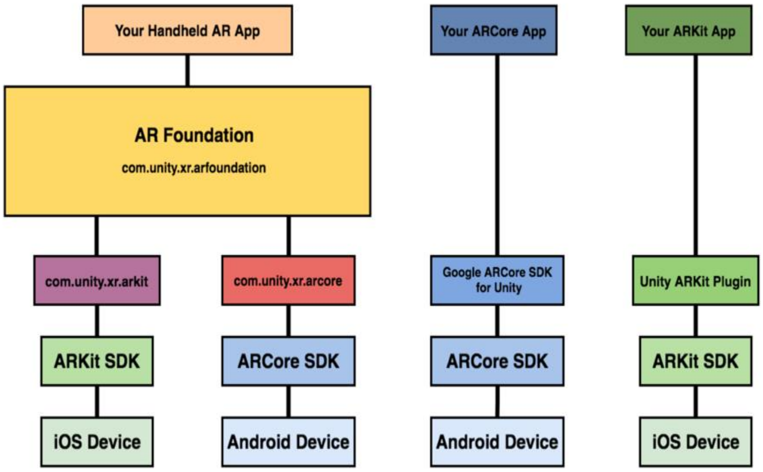

Fig. 2: Support for ARCore and ARKit Platforms of Unity AR Foundation

As shown in Figure 2, Unity's AR Foundation basically provides an abstraction layer to ARCore and ARKit. AR Foundation integrates AR functions shared by ARCore and ARKit, and supports devices running Android 7.0 or higher and iOS 11.0 or higher.

Apple's ARKit framework supports iOS version 11.0 or higher devices. Motion capture is possible by analyzing the screen in real time through machine learning from iOS devices equipped with A12 chip abnormalities (iPhone XR or higher). It also provides an occlusion function that allows virtual objects to be placed between real objects by grasping the depth of objects. Through Scene Geometry technology, it is possible to implement a 3D modeled space by identifying floors, walls, doors, etc. in conjunction with LiDAR(Light Detection and Ranging) sensors. In addition, Location Anchor supports the implementation of specific branches around the world as augmented reality through Apple maps. In addition, there are People Occlusion technology that can grasp the depth of AR contents in the real world, Multiple Face Tracking technology that can apply AR contents such as Snapchat while tracking faces, and Collaborative Sessions technology that can jointly create real-time AR apps with multiple people.

Google's ARCore has been developed as Tango Project's next-generation augmented reality software API(Application Programming Interface). It supports Android 7.0 and later versions, and AR technology can be implemented without additional hardware devices. In addition to poses(Position and Orientation) in the real world, ARCore provides interaction with detected plane and feature points through ray detection. It also provides an anchor that tracks the location of an object. There are motion tracking technology to predict the location of cameras from augmented virtual objects using Plane Detection technology, Light estimation technology to detect the intensity of light in reality, and Environmental understanding technology to identify the real environment. The AR application may change its camera pose due to an update of a continuously changing environment. ARCore's Anchor is a technology that allows virtual objects to stably fix at the same location in the world coordinate system so that they can track the location of virtual objects. Even if the environment is updated through Anchor, the location of the virtual object may be stably maintained.

2.3. Mobile AR Technology with Cloud Technology

ARCore's Cloud Anchor API allows developers to store anchors on the Google Cloud Platform. Through this, developers can create a shareable AR experience. Cloud Anchor uses cloud capabilities to provide ARCore with the same AR service to multiple users(Cao et al., 2020; Manuaba et al., 2021; Morar et al., 2020). Through multi-user access, multiple mobile devices can be used to interact with virtual objects registered in Anchor.

Figure 3 is a flowchart in which virtual objects are created using Cloud Anchor, and other clients share anchor locations through the server to create virtual objects. When a device sends an anchor with peripheral features to a cloud server using the ARCore Cloud Anchor API on Client 1, other devices can request to render the same virtual object simultaneously. When it is confirmed that the anchor sent by client 2 is the same anchor on the cloud server, the virtual object synchronized by client 1 is rendered to the same anchor location. Hosting is mapping a common coordinate system to a given physical space, and when hosting Anchor as a server, it sends visual mapping information(depth and location of the area) related to the user's environment to the Google cloud server. When information is uploaded to the Google server, it is processed as a Sparse point map similar to the point cloud. When the anchor is successfully registered, a unique cloud anchor ID is assigned to the anchor. Resolving is to add a cloud anchor to a scene of devices that has previously hosted a plurality of devices as a physical space.

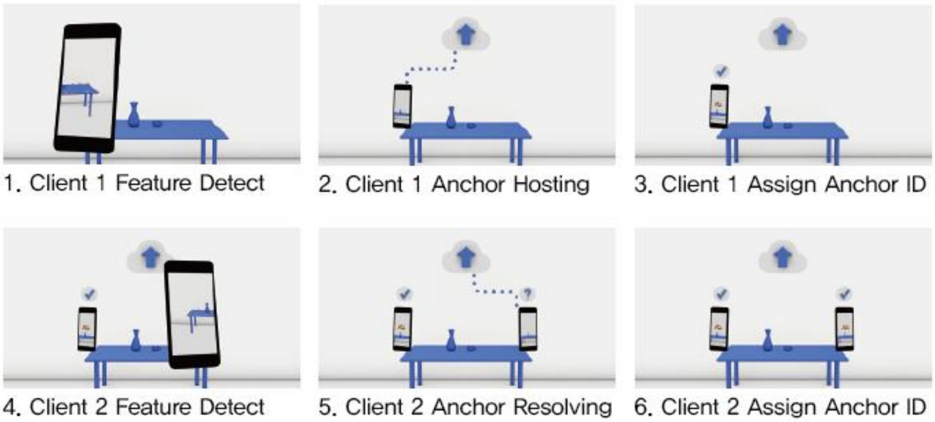

Fig. 3: Google ARCore Cloud Anchor Sequence Chart

3. Coordinate Synchronization System

In this chapter, explains mobile AR theory and Cloud Anchor technology. Through this, it is possible to create a near-field collaboration AR application. However, when a long-distance collaborative AR application needs to be implemented, the same feature points cannot be shared due to different locations. Each client constructs an AR environment map using SLAM. Therefore, each client has a different AR environment, and when a virtual object is created through Anchor, it is created at different viewing locations. This method requires new coordinates to fix the position because the positions at which the virtual objects are created are different. Therefore, to solve this problem, a coordinate synchronization technique using a coordinate synchronization flag is proposed.

3.1. Suggest Coordinate Synchronization Flag

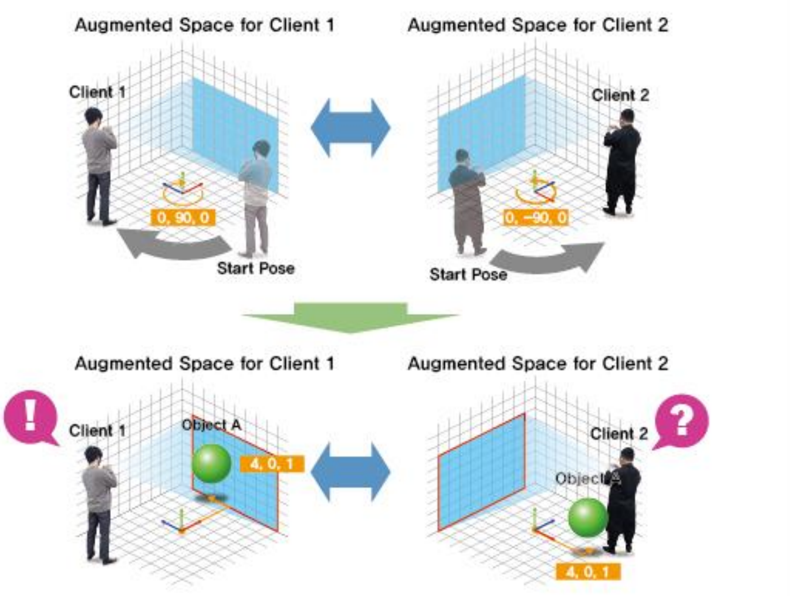

Fig. 4: Example of Not Synchronizing Coordinates

As shown in Figure 4 the world origin for ARCore (e.g. world coordinates 0,0,0) is set based on where the application was turned on or where the mobile device was at the start of the app. Therefore, it is necessary to set the world origin or set it based on the location set in the play space setting step. Each client 1 and 2 has different directions and locations due to the physical environment. Since the world's origin is the same, if you create object A at the location you are looking at, you feel that it has been created at a completely different location.

Figure 5 shows that an object generated after installing a coordinate synchronization flag is generated in a shared coordinate system. When each client installs all coordinate synchronization flags, the origin of the shared coordinate system is placed based on the installed flag. Therefore, all objects generated thereafter perform interaction based on the origin of the shared coordinate system, which is the location of the flag generated in the direction viewed by the user. Since the object generated on the coordinate synchronization flag is generated based on the shared coordinate system, each client can confirm that the object A is generated at the same position in the desired direction.

3.2. Implementing an Object Interaction Network Space

Interaction with an object is to create or move, transform, or delete an object. In this study, canvas models and related painting information were used as objects. The user can create a canvas model by selecting the desired location, and can draw, resize, and delete the canvas model at the desired location based on the generated canvas model. In addition, the user can check information on the current painting situation. Interaction information on these objects is synchronized through the server along with the location and type of interaction. When the user creates a canvas model in the space, flags are installed simultaneously at the corresponding location. The shared coordinate system is created through the flag, and the effects applied to the canvas model are all created based on the shared coordinate system. The transmission priority of interaction was set according to the degree of real-time information. For example, information such as creation or deformation of an object requires a quick response and thus is treated with higher importance than other information.

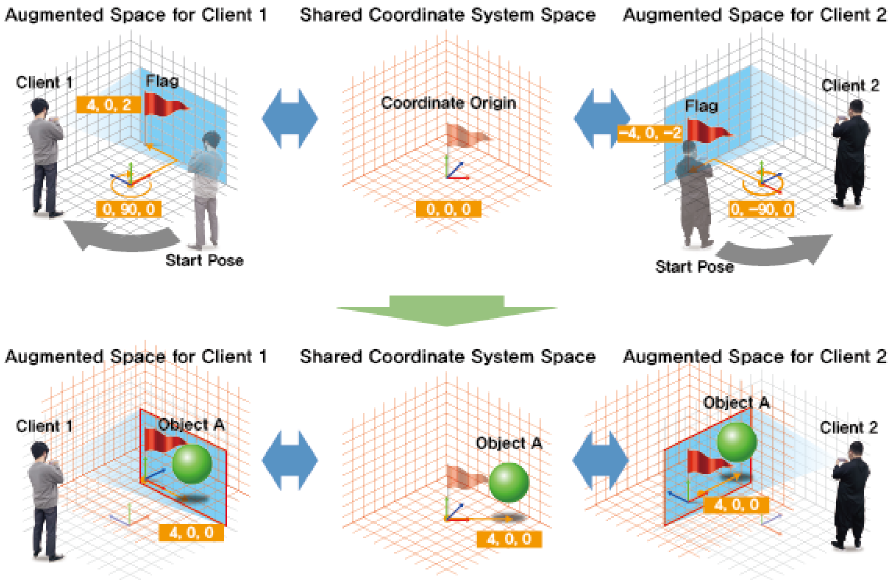

Fig. 5: Example of Using Coordinate Synchronization Flag

The network of collaborative spaces is configured in the form of a server-client through connections between one server and multiple mobile devices. The server performs the role of assigning and broadcasting recognition numbers for each user. When the client accesses the server for the first time and enters a nickname, each user is given a unique ID. The unique ID is used to distinguish the client, and is uploaded to the server along with the interaction information generated by the client and delivered to another client. It also enables the formation of a shared coordinate system on a local client basis. The synchronized painting information is configured in the form of nickname and object information. The interaction information consists of an event value representing the type and a parameter representing additional information. The event is defined by the number of types of interactions to be delivered, and a value is set accordingly. In this study, event values such as the location, type, and color of the painting object were set and used. In this study, the above information was organized into one message so that all clients received the same message through the server. Each client synchronizes the space by making it equally represented in its collaborative space through events and parameters included in the message.

4. Proof-of-Concept Application

In the previous chapter, a method of synchronizing reference points of multiple users as reference points of local clients using coordinate synchronization flags was presented. To verify this, the application is implemented and the proposed method is actually applied through experiments. The painting applications used in each device were built through Unity, and plane tracking and SLAM functions using cameras used the ARCore framework. The network connected data in real time using a PUN2 (Photon Unity Networking 2) server. The experiment consisted of a situation in which three users with three different types of Android devices interact.

4.1. Apply PUN2 to Unity

PUN2 is exported through the asset store to use the Photon server in the Unity development environment. When the PUN2 SDK(Software Development Kit) is applied, various methods and classes provided by Photon can be used. Inherit the MonoBehaviorPunCallbacks Class to create and position LobbyScene and Room that can access servers within Unity. Client synchronization for virtual objects on a network to which PUN2 is applied may be classified into two types according to the frequency of update. The first is to objects that shall be operated continuously updates a different sync method if you can be used. It is used for frequent updates, such as the player's location or the location of objects. For object synchronization, the Photon View component is basically applied to the object to be synchronized. To synchronize with scripts, inherit IPunObservable from Class and implement it using the OnPhotonSerializeView method. PUN2 provides a PhotonTransform View component in advance to interact Transform, Rotation, and Scale in real time according to the movement of the object. However, since it does not automatically solve the dead recovery problem, if you want to compensate for the latency phenomenon, you must implement the script yourself. Second, it is not necessary to continuously update, but when an event is required, an RPC(Remote Procedure Call) synchronization method may be used. For example, it is used when you need to receive an event for behaviors that are not frequently used in the contents. The corresponding synchronization may call the corresponding method from each client by applying [PunRPC] to the method.

4.2. Apply AR Foundation to Unity

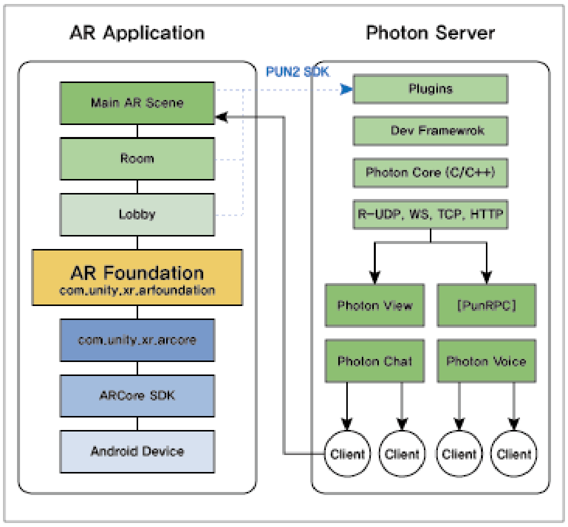

Fig. 6: Flow Chart between AR System and PUN2 Server

To use AR Foundation and ARCore for Unity, Package Manager installs three plug-ins: AR Foundation, ARCore XR Plugin, and XR Plugin Management. ARCore uses an SLAM process called COM(Concurrent Odometry and Mapping) to determine the relative pose of mobile devices in a physical environment. In addition, the pose of the 6-DOF(Degrees of Freedom) is estimated by combining visual information obtained using the feature detection function with the IMU sensor of the mobile device. Through the estimated location, it is possible to render the virtual object from the right perspective by aligning it with the new pose of the virtual camera and AR cameras. Figure 6 is a summary of the data transmission flowchart of the AR Foundation to which PUN2 is applied.

4.3. Build Unity Contents on Android

In order to build contents produced from Unity to ARCore on Android mobile, it changes to the within Unity configuration and mobile device software configuration. The contents does not use a high graphic API for real-time interaction of users, so the OpenGLES3 graphic API is used. Android runs developer mode developer mode and USB debugging mode so that it can be connected to and tested with a real device. In Android, AR technology is available from Android 7.0 or higher versions, so the Minimum API level must be set to 7.0 or higher, and Android software that builds contents also prepares mobile devices with 7.0 or higher versions installed. In this experiment, Samsung Galaxy A90, S20, and Z Flip three devices were used to prove that mobile hardware with different performance interacts equally.

4.4. Running the Application

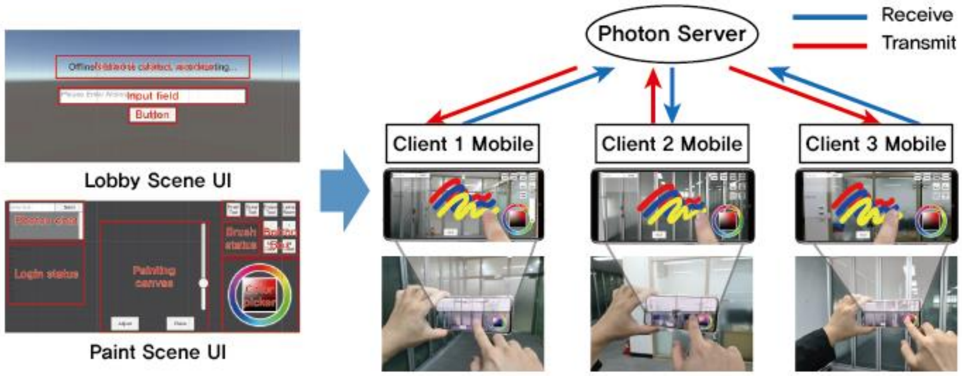

Fig. 7: Implemented Application UI and Data Transmission Method

Figure 7 is a schematic diagram of the data transmission method of the implemented application. Each user uses painting contents using a mobile device in a different space. Each client is connected to the Photon server using PUN2. The interaction event data for the virtual object acting on each client is transmitted to the object of the other client through the server and synchronized to the same state. The contents produced consists of two scenes. The first scene is a space in which each client is prepared to be connected by accessing the server with a Lobby Scene and creating a room. It consists of Text, which informs the current network status, Input field, and Login Buton, which allow you to enter Nickname, which allows you to enter the next scene. Room creation creates a room according to the same AppID and accesses the server. The second scene is a space where each user can proceed with painting contents together with the Paint Scene. If the server connection is successful, plane detection is performed through a mobile camera for AR environment setting. It forms point clusters by finding feature points through images input from ARCore. These surfaces are stored as planes with specific boundaries and can be used to place virtual objects on them.

Once the floor decision is made, the canvas model can be placed on the vertical wall to use the painting contents as shown in Figure 8. The canvas model can be placed where the user wants through ARCore's floor detection, and the size can be freely adjusted. In addition, it is possible to cancel the existing arrangement and place it in the desired location before painting. coordinate synchronization flag is generated from the origin where the canvas model is arranged, and a shared coordinate system is generated. The origin of the ray starts from the touched portion of the mobile display and is projected, and can interact with the virtual object in the shared coordinate system by returning all virtual planes or objects where the ray intersects. Through motion tracking, the pose of the mobile can be found, and the direction of the ray can be known in the virtual world. When light rays are projected onto the canvas model and collision determination occurs, a paint object set at the corresponding position may be generated.

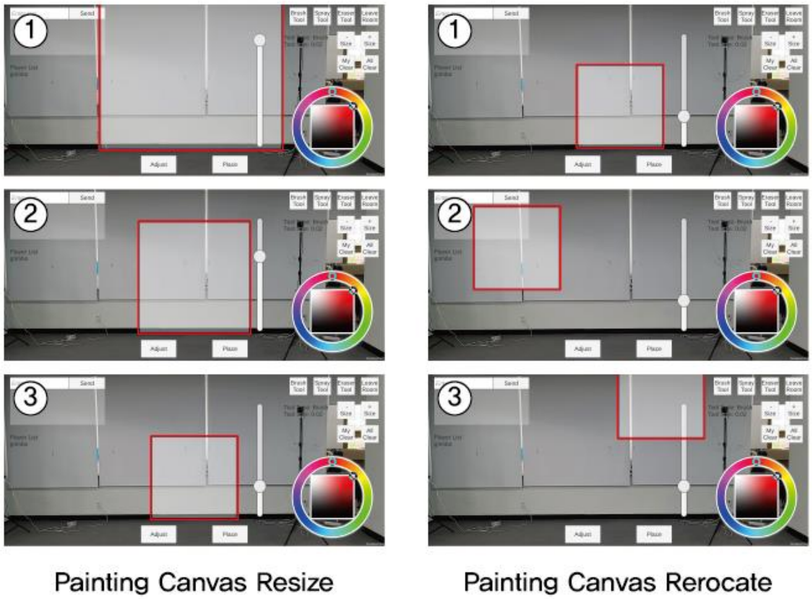

Fig. 7: Implemented Application UI and Data Transmission Method

5. Conclusion

The need for various long-distance application technologies has increased due to the demand for non-face-to-face immersive contents. This paper proposed a technology that synchronizes coordinates so that users can use mobile AR contents from a long distance. The proposed technique is to generate a shared coordinate system based on the coordinate synchronization flag designated by the user. Through this, all events generated by the client occur based on the shared coordinate system. When the interaction of the object of each client occurs, the location is calculated so that the same interaction occurs. This coordinate synchronization can be recognized at the same location because the flag serves as the origin of the reference coordinate system even if each map is generated differently through SLAM. The proposed coordinate synchronization flag allows AR equipment or VR equipment to be used together without distinction. Therefore, the next task will be to apply it to various devices such as smartphones, tablet PCs, and HMD (Head Mounted Display)s. In addition, various follow-up studies will be conducted to increase the interactivity according to each environment by adjusting the rotation and scale of the entire coordinate system using the coordinate synchronization flag.

Acknowledgements

This paper was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.NRF-2020R1F1A1073866).

Also, this paper was supported by the Smart Manufacturing Innovation Leaders program funded by the Ministry of the Trade, Industry & Energy (MOTIE, Korea)

References

Kwok, Andrei OJ, & Sharon GM Koh. (2021). COVID-19 and extended reality (XR). *Current Issues in Tourism*, 24.14, 1935-1940.

Myung-Ha Kim, Sang-Hoon Kim. (2020). Design and Implementation of AR e-commerce System Using Spatial Synthesis Guide Technology. *The Journal of Next-generation Convergence Technology Association*, 4(1), 70-79.

Suh, Ayoung, & Jane Prophet. (2018). The state of immersive technology research: A literature analysis. *Computers in Human Behavior*, 86, 77-90.

Pena-Rios, Anasol et al. (2018). Furthering service 4.0: Harnessing intelligent immersive environments and systems. *IEEE Systems, Man, and Cybernetics Magazine*, 4.1, 20-31.

Ramtohul, Arvind, & Kavi Kumar Khedo. (2022). Location-Based Mobile Augmented Reality Systems: A Systematic Review. *Immersive Technology in Smart Cities*, 41-65.

Dupont, Laurent et al. (2018). A universal framework for systemizing the evaluation of immersive and collaborative performance. *Proceedings of the Virtual Reality International Conference-Laval Virtua,* 1-10.

Ifrim, Andreea-Carmen. (2020). COLLABORATIVE AUGMENTED REALITY– A SURVEY. *The International Scientific Conference eLearning and Software for Education,* 2, 171-178.

Kolivand, Hoshang et al. (2018). Cultural heritage in marker-less augmented reality: A survey. *IntechOpen*.

Pyae, Aung, & Leigh Ellen Potter. (2016). A player engagement model for an augmented reality game: A case of Pokémon go. *Proceedings of the 28th Australian conference on computer-human interaction*. 11-15.

Linowes, Jonathan, & Krystian Babilinski. (2017). Augmented reality for developers: Build practical augmented reality applications with unity, ARCore, ARKit, and Vuforia. *Packt Publishing Ltd.*

Cao, Dongxing. (2020). Real-time multi-user spatial collaboration using ARCore. *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems,* 85-86.

Manuaba, Ida Bagus Kerthyayana. (2021). Mobile based Augmented Reality Application Prototype for Remote Collaboration Scenario Using ARCore Cloud Anchor. *Procedia Computer Science*, 179, 289-296.

Morar, Anca et al. (2020). Evaluation of the ARCore Indoor Localization Technology. *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet), IEEE,* 1-5.