Interactive Shared Mixed Reality on Mobile Devices

Iftekhar Mahmud Towhid

tlabs.im@gmail.com

Department of Computer Science and Telecommunication Engineering,

Noakhali Science and Technology University, Noakhali-3814, Bangladesh

Abstract

Popularity of augmented reality games like 'Pokémon Go' has proved that technologies like virtual and augmented reality are ready for the big time. There has been a number of augmented/mixed reality apps for mobile platforms where a user's experience is generally isolated or individual. In this paper, we discuss about the concept of interactive shared mixed reality using mobile devices where multiple users can share the same augmented experience from their own devices. We wish to demonstrate the concept by means of a multiplayer mixed reality game. Shared mixed reality on mobile devices has specific challenges which are discussed in this paper. Besides, in order to provide unified experience for all the participating users, the system requires special capabilities and protocols. In this paper, we are proposing a framework for shared mixed reality using mobile devices. The purpose of this framework is to facilitate development of similar interactive shared mixed reality applications for mobile devices.

Keywords: Augmented Reality, Mixed Reality, Shared Mixed Reality, Mobile AR

1 INTRODUCTION

1.1 Background

There has been major shifts in how we interact with technology in the recent decades. First personal computing reached our homes in the 80s and 90s to make computers more accessible to general people. Then came the mobile phones which changed the way of our communication. Smartphones might the seen as the most influential piece of technology which changed the whole landscape of our interaction with technology. Smartphones are now the center piece of our digital life and its application has be diversified beyond just communication and entertainment. So, when it is asked, what will drive the next major shift in technology, many would suggest virtual and augmented reality as the most competent candidate.

Both virtual and augmented reality intends to blend our real and digital life, albeit in different ways. Virtual reality, experienced normally with a head mounted display device (HMD), obstructs our whole vision to display virtual objects which can be interacted with using sensors and gesture inputs. In contrast, augmented reality (also sometimes called mixed reality) projects virtual objects in a less immersive way because the users can still view his/her surroundings. The virtual objects can be blended with real objects to create a more realistic experience using complex hardware and software techniques, or they might be rendered simply as an extra layer over real vision.

Augmented reality or mixed reality is definitely a hot new interface for interacting with our digital life. Though the major application of VR and AR is thought to be content consumption and gaming, it can also be used in many other fields. These technologies already painted their footprints in the field or remote collaboration [1-3], construction and repairing, design, prototyping and architecture [4], education [5] and entertainment; and the list is growing.

In the next sections of this chapter, we are going to discuss about the state of the art of VR and AR. Then we are going to discuss a new concept of how to extend the experience by a new technology called Shared Mixed Reality (SMR), which we wish to implement on mobile devices as a low cost and ready to deploy solution.

1.2 State of the art

There has been a slew of virtual reality (VR), augmented reality (AR) and mixed reality (MR) devices for the last few years. For example Oculus Rift [6], HTC Vive [7], PlayStation VR [8], Samsung Gear VR [9], Google Cardboard VR [10] are popular hardware which offer VR experiences with various degree of immersiveness and fidelity. Devices like Google Glass [11], Meta [12] and Microsoft Hololens [13] provide augmented and mixed reality experience which is technologically much more complex than VR.

Microsoft Hololens is the most technologically advanced mixed reality device, which is currently only available for developers and costs a lot. Generally most VR and AR devices with good fidelity and immersiveness are pricey enough to be considered premium devices and many also require powerful peripherals like PC [14]. Luckily, devices like Samsung Gear VR and Google Cardboard VR present us a proof that smartphones are now powerful enough to provide an acceptable VR experience. On the AR/MR front in mobile platforms, there are a number of AR apps available which provide a limited to moderate level of AR/MR experience [15]. For example, 'Pokemon Go', an AR application if one of the most grossing app in the recent year, though the implementation of AR in the app is really basic.

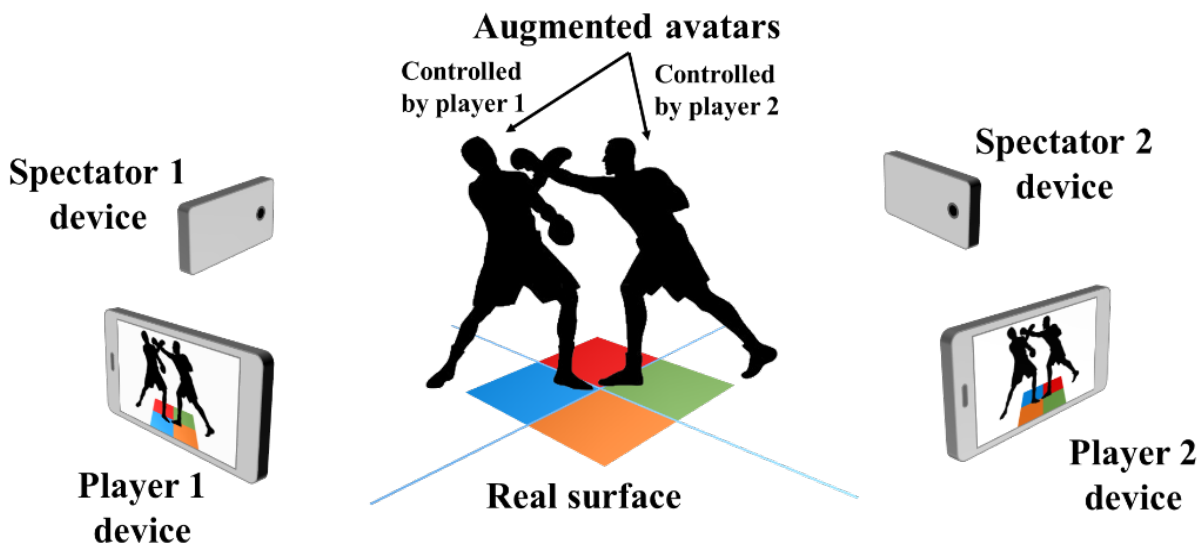


Figure 1: An example of interactive shared mixed reality using smartphones. Two avatars controlled by two players may engage in fighting, where the players see the avatars augmented/mixed with real world through their smartphone cameras. Spectators may join with their smartphones to watch the fighting.

The motivation of this research comes from two simple observations. First, mobile devices like smartphones can now provide decent VR/AR experience and second, the VR and AR experience we get from mobile devices through various apps are completely personal or individual, which means, only one user enjoys the experience using the mobile screen. User uses his/her devices to view and interact with augmented objects. These augmented experiences can only be shared by sharing the device screen. In case of interactive augmented objects, a single user only controls the interaction. More advanced MR devices like Microsoft Hololens provides the feature to share one's experience with others [16]. Hololens is equipped with advanced spatial tracking technology, which facilitates the sharing. Also, recently Apple introduced similar capabilities into their mobile devices, but they are only limited to Apple platforms.

This made us think – what if mobile VR/AR experience could be shared? It is soon realized that, AR or MR, is more suitable than VR for that case. We can think of a few use cases where shared mixed reality can provide unique and a whole new level of experience. For example, a shared mixed reality dual fighting game, or other multiplayer games. This also can open a new door in collaborative learning is educational institutions.

1.3 Basic Definitions of related technologies

1.3.1 Virtual Reality (VR)

The Virtual reality (VR), which can be referred to as immersive multimedia or computer-simulated reality, replicates an environment that simulates a physical presence in places in an imagined world, allowing the user to interact in that world.

1.3.2 Augmented Reality (AR)

The Augmented reality (AR) is a live, direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data.

1.3.3 Mixed Reality (MR)

The Mixed reality (MR)—sometimes referred to as hybrid reality—is the merging of real and virtual worlds to produce new environments and visualizations where physical and digital objects co-exist and interact in real time.

Hardware related with mixed reality includes Microsoft's Hololens, which is set to be big in MR—although Microsoft have avoided the AR/MR dispute by presenting yet another term: "Holographic computing".

1.4 Objectives of Research

A common problem, current state of VR/AR/MR suffers is fragmentation. Most of the available VR/AR devices come with their own platform and are incompatible with one another. To our knowledge, shared mixed reality on mobile devices is a completely new field of research, though Microsoft Hololens provides the ability to share experiences. In this paper, we are presenting a framework for the next iteration of the MR experience– shared mixed reality on mobile devices.

The motivation for the framework comes from the challenges faced during the making of a mixed reality game. The game enables two in-game avatars fighting each other that are controlled by two human players from their own devices. This is like an online multiplayer game, but the difference is – instead of viewing the game objects on the virtual environment on the screen, the game objects are augmented to the real surroundings as viewed by the camera sensors of each player's mobile devices.

The two players see the same augmented objects but from different perspectives as shown in figure 1. The players control their avatars to fight against the opponent, so the transformations of game objects need to be properly synchronized for making sure that the players see the exact same thing. Another feature we wanted was to share the game activities beyond the players.

Any other person can join and see what the players are doing in the game. These spectators do not interact in the game but their mobile devices may be used to enhance the experience.

The objective of the framework is to facilitate subsequent development of similar SMR application. Creating such a framework involves overcoming a number of challenges which stems from the current limitation of mobile devices. Our end goal is to develop such a framework which can leverage these limitations to provide an acceptable experience with good fidelity and immersiveness.

2 PROPOSED FRAMEWORK

Shared mixed reality can be compared to online multiplayer games. In fact, this is where we are taking inspiration from. In an online multiplayer game, players interact with the same environment, from their perspective and the experience in unified and synchronized, i.e. all the players see the same thing, albeit from different perspective. This is also our first requirement in creating the framework – to make the experience uniform for all players and spectators. Thus we choose a centralized model rather than a distributed one. So our proposed model incorporates a server which may reside on a dedicated machine or one of the player's device.

There are two major steps for playing a SMR game:

1. Matchmaking
2. Match playing

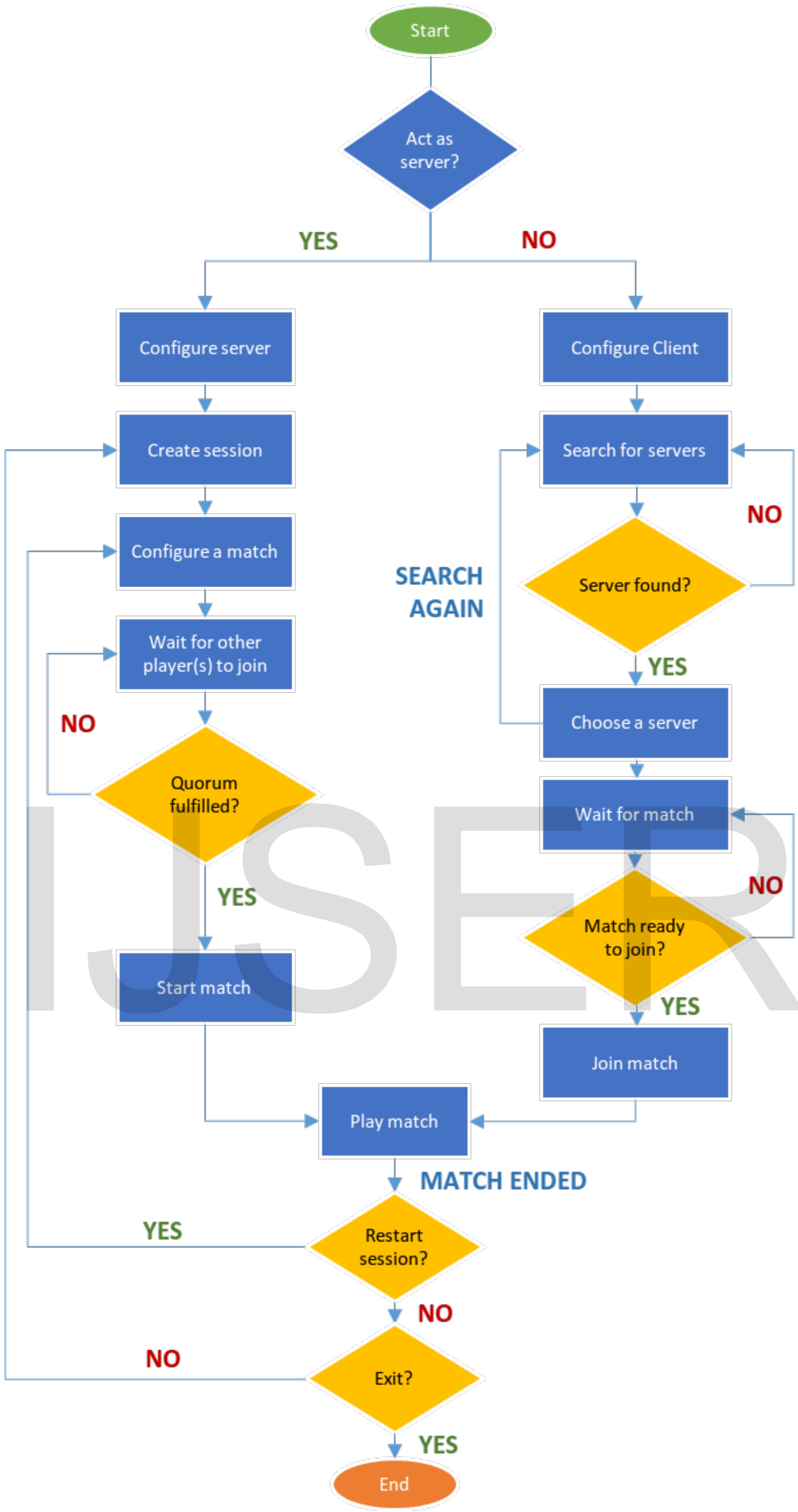


Figure 2: Matchmaking and match proceedings

2.1 Matchmaking

Matchmaking allows players to create match or discover existing matches over the network. At startup, there are two options for a player. For one, he can create a server and then configure session and match so that other players can join in. If the player opt to act as a client instead, the app will search in the network for configured matches which are waiting for other players to join. Upon discovering a match on the network (presumably a wireless network), the players connect to the server as clients.

The process in described in details in figure 2 which shows the flow diagram to matchmaking process. The server hosts the matches under a session. This allow other players to remain connected even after a match ends. On agreement, the players might choose to play another match during the same session.

The client might find more than one hosting servers with a session initiated. Among them, the client is connected to one at a time. After joining, the client sees the already configured match or wait for the match configuration to be finished. When the match is properly configured, the player might choose to join the match id there are places available. The matches might a to-player or multiplayer game. So after the quorum is fulfilled the server will start the match.

2.2 Match playing

Figure 3 shows our proposed mobile SMR framework. To simplify the framework, the system requires that players only send their inputs (touch events) to the server. This helps in reducing the overall traffic over the network, as in practice, other player properties are of no use to the server. Player inputs are scrutinized and filtered to determine which input events are to be processed. This step may be optional for some applications, but is crucial for games where timing of player input events are critical in determining the game state. For example, in gun-fighting games, it may be crucial to know which player had pressed the trigger first in close combat situation.

The server contains a global 3D world definition file which is used to store the definition of the 3D world that is augmented to the players' perspective. Each player also contains a local copy of the file. The global 3D world definition file is modified based on the permitted player inputs. A differential update generator determines which part of the definition of the world is transformed or updated, and creates a differential update which is to be transmitted back to the players' devices. We propose that the 3D world definition file system can differ from application to application, so our system is file-system independent. The file-system should be optimized such that the differential updates can be generated and are small in size.

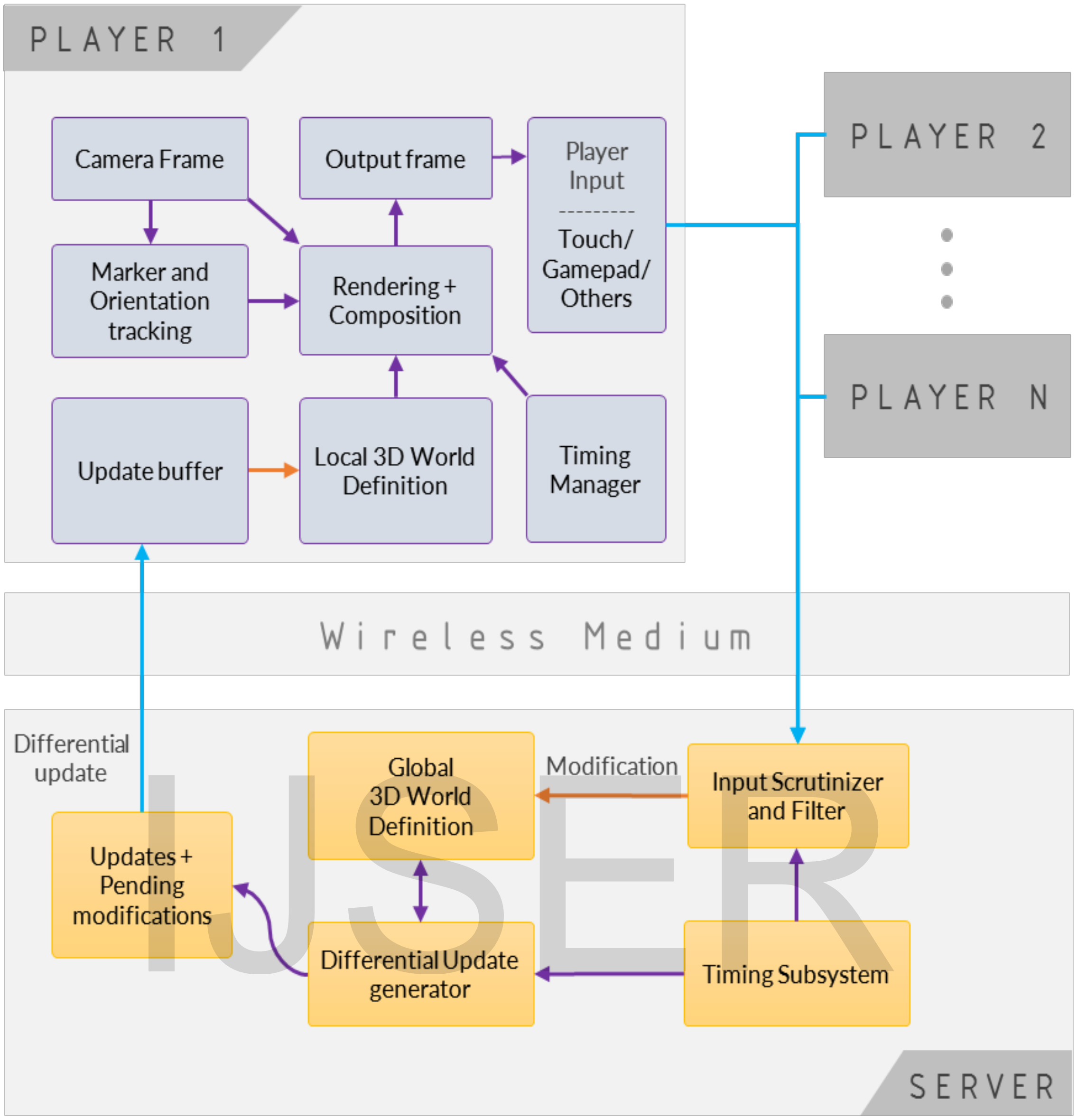


Figure 3: Proposed Mobile SMR Framework

Based on the differential update received, the players update their local 3D world definition file. At each step, we take a frame of the real world, captured using the viewer's smartphone camera, identify the distance and orientation of the marker in the frame, render the 3D world based on these and along with other camera properties such as focal ratio, and finally composite the rendering with the actual frame. Other advanced methods can be used to make the composition more realistic such as shadow casting by light-source detection, but they might reduce the performance in lower spec devices. To simplify the framework, but provide a general experience for all players, we propose the frame rates be locked at 20/25/30 fps to provide varying level of experience for devices with the varying specification.

```
smr_pong: {
    player1 : {
      paddle_x: 0,
      score: 0
    },
    player2: {
      paddle_x: 0,
      score: 0
    },
    ball: {
        state : {
            position: {
                x: 0,
                y: 0
            },
            orientation: {
                x: 0,
                y: 0,
                z: 0
            }
        },
        direction:{
            x: 0,
            y: 0
        },
        speed: 0,
        acceleration: 0
    }
}
```

Figure 4: An example of a simple world definition file for our demo PONG game. Here the world state is defined as a JSON file. The obvious advantage of such file format is, it is light-weight and easier to generate differential updates, yet enough for each player and spectators to render the virtual world.

The output frame is shown on the player's device screen and then based on the output, the players may interact by proving input feedback likely by touching the onscreen objects and controls. These input events are then forwarded to the server.

The spectators' experiences are similar to that of the players' except they do not interact with the virtual environment.

3 Designing a SMR application

3.1 Challenges

Designing an SMR application has specific challenges to be met. An SMR application comprises of four sub-systems, these are -

1. Mixed Reality algorithm
2. Networking
3. Match Controller
4. Game/App Logic

These four sub-systems should work in tandem to make a SMR application work seamlessly across devices.

3.2 Mixed Reality

Though Augmented Reality and Mixed reality seem very similar in principle, in mixed reality, the virtual objects are placed such that they seem to be a part of the real environment through the visor (i.e. the mobile screen), rather than an overlay as it is in augmented reality. To enable this, there need to be some means to map the environment in which the virtual objects are to be placed. There are two main approaches for such augmentation, marker-based and marker-less. In marker-based AR an object - most commonly a 2D image printed on paper is used as marker, which is used as a reference for render augmented objects. In contrast, marker-less AR is more complex and often requires specialized hardware like dual camera, specialized camera, Infrared and Ultrasonic sensors etc. The main goal here is to make a 3D blueprint of the surrounding environment, where objects are manually pinned. Marker-less AR has more challenges than marker based AR and is costlier to develop. In this paper, we will concentrate mainly on marker-based AR, as it should work in any mobile devices without requiring special hardware other than the rear camera. It should be noted that for devices with single rear camera, marker-less AR will fail when the surface on which the objects are to be placed, lacks distinctly rich and asymmetric textures.

3.2.1 Marker based Mixed Reality

Markers are used as reference points for tracking the relative position between stationary and moving objects. For example, in production of movies, motion tracking markers helps to capture tracking data for video editing and post production. In mixed reality applications, generally a single printed image in used as a marker, which contains a lot of feature points.

When choosing an image to use as a marker, there are a few simple things to bear in mind to ensure good performance. The marker tracking performs best when used on rich and highly textured images, so it's better to have a marker containing lots of details, rather than bold lines or blocky designs. The latter will have less detectable local regions so that detection will suffer.

ISSN 2229-5518

Using the tracked feature points, a 3D reference place in constructed, in reference to which the virtual objects are placed. A fine advantage of such marker-based tracking is, it works regardless the focal ratio of the camera lens.

There are lot of open-source and proprietary AR/MR library/SDK available, i.e. Vuforia [17] which, when used with a good marker, works really well in practice. A few SDKs such as CloudRidAR [18] leverage cloud computing for performance improvement. Other example of existing AR SDKs are ARToolKit [19], Catchoom CraftAR [20], Mobinett AR [21], Wikitude [22], Blippar [23], Layar [24], Meta[12] and ARLab.

A good tracking algorithm should possess some important capabilities. First, it should be fast enough to make the tracking smoother. A slow tracking algorithm will introduce jitter, which may ruin the experience. Second, it should work even when the marker is partially obscured. Advanced algorithms may include occlusion detection and light source detection capabilities to make the augmentation more realistic.

3.3 AR/MR algorithms

A fundamental measure of AR systems is how convincingly they renders augmentations with the real world. The software must interpolate real world coordinates, independent from the camera, from camera images. This process is known as image registration, and uses different methods of computer vision, mostly related to video tracking [25][26], A lot of computer vision methods for AR are acquired from visual odometry. Typically those methods consist of two stages. The first stage is to identify interest points, fiducial markers or optical flow in the camera images. This step can use feature detection methods like corner detection, blob detection, edge detection or thresholding, and other image processing methods [27][28]. The second stage restores a real world coordinate system from the data obtained in the first stage. Some methods assume objects with known geometry (or fiducial markers) [29] are present in the scene. In some of those cases the scene 3D structure should be pre-calculated earlier. If part of the scene is obscured, simultaneous localization and mapping (SLAM) can map relative positions. If no information about scene geometry could not be retrieved, structure from motion methods like bundle adjustment are applied. Mathematical methods used in the second stage include projective (epipolar) geometry, geometric algebra, rotation representation with exponential map, kalman and particle filters, nonlinear optimization, robust statistics.

Augmented Reality Markup Language (ARML) is a data standard developed within the Open Geospatial Consortium (OGC) [30,] which consists of XML grammar to describe the location and appearance of virtual objects in the scene, as well as ECMAScript bindings to allow dynamic access to properties of virtual objects.

3.2 Networking

Networking on mobile devices is the apparent pain point in designing a multiplayer system. Pinging in perceivably slower and the latency is inconsistent in mobile devices. The performance heavily depends on the quality of wireless network, and using high performance routers and networking chips on mobile devices, the overall performance can be improved. Regardless, as SMR applications require real-time synchronization among multiple devices, the networking codes should be well optimized. It should be noted that, the other three sub-systems, namely, mixed reality, match controller and game/app logic codes depends heavily on the networking subsystems.

There are a few solutions available for implementing local networking in mobile devices. For implementing our demo game we used a combination of custom networking library written in C# and the built-in networking library of Unity3D.

3.3 Match Controller and Game/App logic

The match controller is responsible for managing matches. A server can create a match, which should be discoverable in the network, where the players and spectators can join. The match controller controls the creation, scoring and termination of matches. The game or application logic depends on the type of game/application and the developer is responsible for the implementation.

4 DEMONSTRATION OF THE FRAMEWORK

4.1 SMR PONG Game

To demonstrate our proposed architecture, we chose to build a shared mixed reality PONG game. A PONG is a tennis-like two player game generally played on table. This game logic is relatively simple, but it is a perfect example for demonstrating the power and possibilities of shared mixed reality.

The game was developed using Unity game engine. For AR part, we used Vuforia library which handles all the MR related jobs like marker detection and rendering. For networking, we used the combination of Unity's built in networking capabilities and our custom networking library.

The following figures shows some in-app screenshots of our game, running on android phones. Here one device is acting as a server and hosting a match, and other device has joined as client player. The two devices are placed at the opposite ends of the marker, so the players essentially see the same virtual objects, though from different viewpoints.

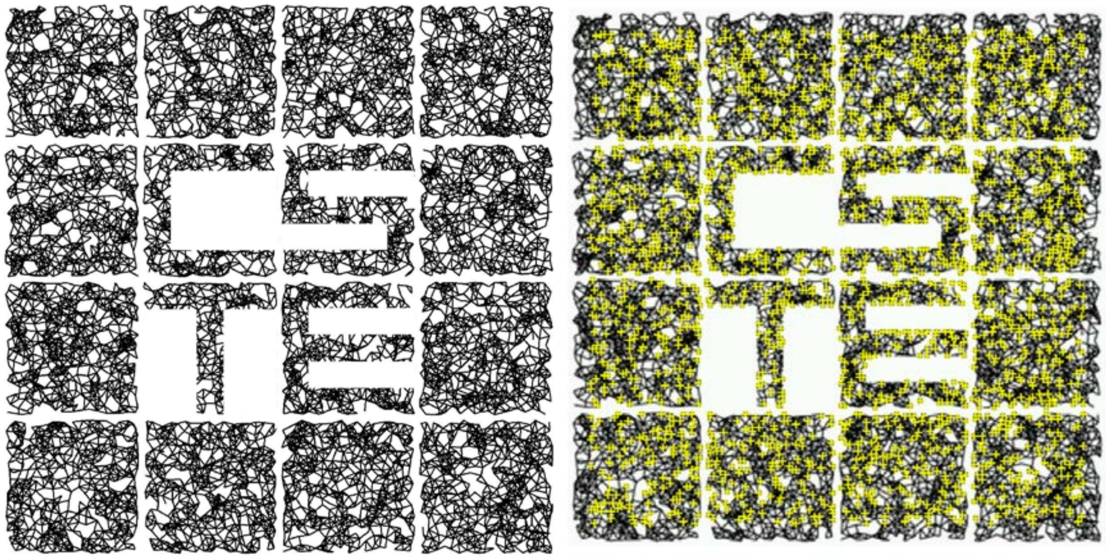
Figure 5: In-game screenshot- View from host player's mobile device

Figure 6: In-game screenshot- View from client player's mobile device

4.2 Marker Design

While designing the marker for our application, we considered a few things - first the marker should be easily printed on any kind of printer, so we chose to design a monochrome marker. Our marker is designed such that, it can be easily printed on an A4-sized paper using any black and white printer. Second, to increase the number of feature points, rather than using any photograph as the marker, we wrote a simple script in C# to auto generate the marker with a lot of uniformly distributed feature points.

Figure 7 shows the marker used in our demo game. The marker has enough feature-points as shown in Figure 7-b and can be detected even if some part of it is obscured. The asymmetry of the marker allows to continuous detection while moving 360 degree around the marker.



(a) (b)

Figure 7: a) The marker was auto-generated using a C# script b) The feature points of the marker

5 CONCLUSION

5.1 Conclusion

Shared mixed reality is a brand new space for research and it has the potential to revolutionize mixed reality experiences using our existing day to day devices like smartphones. Apart for gaming and other entertainment applications it can be used in other collaborative fields, such as education and productivity. In particular, interactive SMR in the field of education can transform the collaborative learning in primary and higher levels and make the learning more enjoyable and entertaining. For example, an app can be thought of, where a group of kids are tasked with sorting virtual letter blocks collaboratively using the mobile and tablets as the SMR device. A SMR framework can incentivize and streamline subsequent development of such applications and prevent fragmentation is this new field.

Our goal is to make the framework robust enough to work with devices with varying capabilities and specifications. The framework is simple yet effective, in the sense that it can be implemented in a wide range of devices without requiring specialized hardware, though specialized hardware can improved the experience further.

There is no other similar open source frameworks available to compare, so we were unable to compare the performance and efficiency of our framework.

4.2 Future work

Available marker detection algorithms works quite well in practice, but we wish to develop our own implementation for marker detection, with the goal to optimize the system for our custom marker and thus reduce the latency. Furthermore, we wish to detect the ambient lighting of the environment and direction of available light sources, so that the augmented objects can be rendered with more realistic lighting and shadows. We also wish to optimize the networking library to reduce network latency to improve overall quality of gameplay. Finally, we wish to implement a more complex game to further investigate the performance and possibilities of our framework.

REFERENCES

[1] H. Regenbrecht (2004), 'Using augmented virtuality for remote collaboration'. Presence: Teleoper. Virtual Environments 13(3) pages 338-354

[2] 'Effects of HoloLens in Collaboration: A Case in Navigation Task's Eswara Rao, Velamkayala, Manuel V. Zambrano, and Huiyang Li, Proceedings of the Human Factors and Ergonomics Society Annual Meeting. https://doi.org/10.1177/1541931213602009

[3] Matthew G. Hanna, Ishtiaque Ahmed, Jeffrey Nine, Shyam Prajapati, and Liron Pantanowitz (2018) Augmented Reality Technology Using Microsoft HoloLens in Anatomic Pathology. Archives of Pathology & Laboratory Medicine: May 2018, Vol. 142, No. 5, pp. 638-644. https://doi.org/10.5858/arpa.2017-0189-OA

[4] https://www.ife.no/en/ife/departments/software-engineering/files/ar/ar-in-architectural-and-urban-planning.pdf

[5] https://medium.com/inborn-experience/usage-of-augmented-reality-in-education-be783e0159a

[6] https://www.oculus.com/rift/

[7] https://www.vive.com/

[8] https://www.playstation.com/en-ae/explore/playstation-vr/

[9] https://www.oculus.com/gear-vr/

[10] https://vr.google.com/cardboard/

[11] https://www.x.company/glass/

[12] www.metavision.com/

[13] https://www.microsoft.com/en-us/hololens

[14] https://www.vive.com/eu/support/vive/category_howto/system-requirements.html

[15] www.pokemongo.com/

[16] https://developer.microsoft.com/en-us/windows/holographic/shared-_holographic_experiences

[17] https://developer.vuforia.com

[18] Huang, Z., Li, W., Hui, P., Peylo, C. [1]. CloudRidAR: A Cloud-based Architecture for Mobile Augmented Reality Proceeding of MARS'14, July 2014.

[19] AR ToolKit, available at https://github.com/artoolkit

[20] http://catchoom.com/product/craftar/augmented-reality-sdk/

[21] http://www.mobinett.com/

[22] http://www.wikitude.com/

[23] https://blippar.com/

[24] https://www.layar.com/

[25] Azuma, Ronald; Balliot, Yohan; Behringer, Reinhold; Feiner, Steven; Julier, Simon; MacIntyre, Blair. Recent Advances in Augmented Reality Computers & Graphics, November 2001.

[26] Maida, James; Bowen, Charles; Montpool, Andrew; Pace, John. Dynamic registration correction in augmented-reality systems Archived 18 May 2013 at the Wayback Machine., Space Life Sciences, NASA.

[27] State, Andrei; Hirota, Gentaro; Chen,David T; Garrett, William; Livingston, Mark. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking, Department of Computer ScienceUniversity of North Carolina at Chapel Hill.

[28] Bajura, Michael; Neumann, Ulrich. Dynamic Registration Correction in Augmented-Reality Systems University of North Carolina, University of Southern California.

[29] Fiala, M. (2005) 'Artag, A Fiducial Marker System using Digital Techniques', in IEEE Proceedings on Computer Vision and Pattern Recognition (San Diego, June, 2005), CVPR'05, Volume 2, IEEE Computer Society, pages 590- 596

[30] "ARML 2.0 SWG". Open Geospatial Consortium website. Open Geospatial Consortium. Retrieved 12 November 2013.

[31] Open Scene Graph, available at http://www.openscenegraph.org

[32] https://appreal-vr.com/blog/markerless-vs-marker-based-augmented-reality/

[33] http://studierstube.icg.tugraz.at/thesis/marker_detection.pdf

[34] https://wiki.kudan.eu/What_Makes_a_Good_Marker