A Reality Check of Positioning in Multiuser Mobile Augmented Reality: Measurement and Analysis

Na Wang
George Mason University
Fairfax, VA, USA
nwang4@gmu.edu

Haoliang Wang
Stefano Petrangeli
Viswanathan Swaminathan
Adobe Research
San Jose, CA, USA
{hawang,petrange,vishy}@adobe.com

Fei Li
Songqing Chen
George Mason University
Fairfax, VA, USA
{fli4,sqchen}@gmu.edu

ABSTRACT

Multiuser Augmented Reality (MuAR) is essential to implementing the vision of Metaverse. With the pervasive mobile devices, MuAR enables multiple devices to share a common AR experience. In such experiences, the peer positions are critical to understand peers' intentions and actions so as to achieve the smooth interaction in AR. Such a spacial awareness requirement poses new challenges to MuAR. Traditionally, in AR experiences designed for the single user, the SLAM algorithm is adopted to compute self positions. However, the computed positions cannot be directly used to compute the relative positions of peer devices in MuAR, because they are computed with respect to independent coordinate systems associated with participating devices. To fill in the gap, the industry has recently proposed to implement peer tracking with the help of built-in Ultra Wideband (UWB) chip. In this work, we aim to perform a reality check on the proposed support, with the Nearby Interaction (NI) framework developed for iOS mobile devices as an example. The goal of our study is to gain an in-depth understanding about the reliability of the proposed support and identify potential issues. Through extensive measurements, we discover the peer tracking solution is not reliable sometimes, in terms of availability and accuracy. Furthermore, with regard to erroneous position reports, we present a quantitative analysis, summarizing the error types (e.g., transient errors and permanent errors) and revealing their underlying reasons. We believe the preliminary findings could help to improve the spacial awareness and enhance user experiences in MuAR.

KEYWORDS

Augmented Reality, Multiuser AR, Spatial Awareness, Tracking

ACM Reference Format:

Na Wang, Haoliang Wang, Stefano Petrangeli, Viswanathan Swaminathan, Fei Li, and Songqing Chen. 2022. A Reality Check of Positioning in Multiuser Mobile Augmented Reality: Measurement and Analysis. In *ACM Multimedia Asia (MMAsia '22), December 13–16, 2022, Tokyo, Japan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3551626.3564966

MMAsia '22, December 13–16, 2022, Tokyo, Japan

ACM ISBN 978-1-4503-9478-9/22/12.
https://doi.org/10.1145/3551626.3564966

1 INTRODUCTION

Augmented Reality (AR) is considered a foundational building block for Metaverse [25, 20, 27]. With the pervasive adoption of mobile devices, it is expected that much of future AR experiences will run on mobile devices [26, 15]. AR overlays virtual objects onto the real world, allowing users to explore the augmented world in six degrees of freedom (6DOF), providing immersive experiences in various areas [19, 22]. It is reported that the AR market is expected to grow to $ 225.77 billion by 2026 [24]. Furthermore, with the introduction of new AR development toolkits [3, 16, 9, 10, 4], and hardware advances in mobile devices [8], AR gradually evolves to support multiuser experiences [11, 18], in which multiple devices share a common experience so as to implement the collaboration.

To enable multiuser AR (MuAR), it is necessary for participating devices to periodically share peer positions since the information often directly reflects peers' intentions and actions. Moreover, the knowledge of peers' moves is critical for the smooth interaction in MuAR [10, 18]. In the single user experience, popular AR frameworks, such as Apple ARKit and Google ARCore [3, 16], provide the access to the explicitly real-time position and orientation (known as the pose) of mobile devices [1, 17]. The pose is computed by the algorithm of Simultaneous Localization and Mapping (SLAM) [21]. However, the computed positions cannot be directly used to compute relative positions of peer devices in MuAR. The reason lies in that, in the single user case, the coordinate system of the AR world is constructed with respect to the initial pose of the specific mobile device [2]. In other words, the computed positions by SLAM are not relative to peers.

To address the problem of peer tracking in MuAR, the industry has proposed to utilize Ultra Wideband (UWB) chips, recently introduced to iOS devices [12], to provide real-time relative positions in MuAR. More specifically, the high-frequency capability of the UWB chip is utilized for the devices communication [23]. Therefore, during the experience, it is required for peer devices to be in proximity and always present in the field of view (FOV) of each other. The requirements restrict users' movement in the experience, and thus, may conflict with the experience design of developers, leading to the degraded user experiences.

In order to provide a realistic and quantitative understanding of such a support to AR users and application developers, we, for the first time, perform a measurement study about its reliability, focusing on the Nearby Interaction framework developed by Apple Inc. Through extensive experiments, we find the framework may produce incomplete and/or inaccurate position reports. With regard

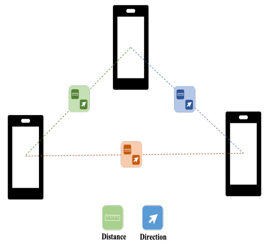


Figure 1: Peer Tracking by Nearby Interaction

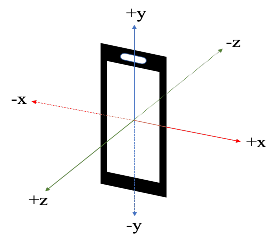


Figure 2: The Coordinate System in AR

to erroneous position reports, we present a further analysis summarizing characteristics of errors (e.g., transient errors and permanent errors) and identifying the underlying reasons. The highlights of our findings are as follows:

- The existence of obstacles in the line of sight of peer devices has the greatest influence on the peer tracking in MuAR: the relative distance reports are not accurate while the relative direction reports are entirely unavailable.
- The orientation of device screens in the experience mainly impacts the availability of the relative direction reports.
- The distance between devices has the least impact on tracking: the accurate position reports are always available, until the distance exceeds 20 meters.

The rest of the paper is organized as follows. Sec. 2 presents the background on related industry solutions. Sec. 3 details our measurement findings. Sec. 4 analyzes the measurement errors qualitatively and quantitatively. The paper is concluded in Sec. 5.

2 BACKGROUND

The Nearby Interaction framework is the first industry solution to provide the peer tracking in MuAR [10]. It works on those iOS devices equipped with Ultra Wideband (UWB) chips. The framework provides two measurements for the peer tracking: the relative distance and relative direction, as shown in Figure 1.

The relative distance is measured from one device to its peer in meters while the direction is a vector pointing from one device in the direction of its peer, with the x-axis extending positively to the right, y-axis extending positively upward, and z-axis extending negatively from the device's center, away from the user, as illustrated in Figure 2 [6, 5]. Both measurements are jointly used to determine the one and only relative position of the peer device.

3 MEASUREMENT STUDY

The Nearby Interaction (NI) framework enables multiple users to share a common AR experience. However, NI only provides complete and accurate position reports of peer devices when a series of requirements are met, as illustrated in Figure 4 [7]. *First*, participating devices should be in close proximity. *Second*, the screens of devices should be kept in the portrait mode. *Third*, devices should always appear within the line of sight of each other and no obstacles are present within the line. In practice, the requirements would restrict users' movement freedom, and thus conflict with the experience design, leading to the degraded user experiences.

3.1 Experiment Setup

To study how the peer tracking can be impacted by violations of aforementioned requirements, we perform a series of measurements. Without the loss of generality, we focus on the two-device scenario in which one device is fixed while the other one moves around. For the remainder of the paper, we refer to them as Device A and Device B, respectively. Besides, the screen of Device A is always locked into the portrait mode as required. Since the framework is able to simultaneously support several devices, the conclusions here also apply to scenarios involving more devices.

Also, to evaluate the accuracy of the framework reports, we employ the Ultra-Wideband (CUWB) system by Ciholas for the collection of ground truth positions in experiments [14]. The system uses the wide bandwidth to achieve the real-time tracking. The system is adopted for its high accuracy, affordable cost and easy setup. The CUWB system can be composed of multiple anchors and tags. The anchors work as static UWB transceivers, providing reference locations. All tags, bound to mobile devices, are tracked in real time. In our experiments, we use four DWETH111 anchors, shown in Figure 5 and two DWTAG100 tags. The positions of tags (same as mobile devices) are transmitted as User Datagram Protocol (UDP) packets [13].

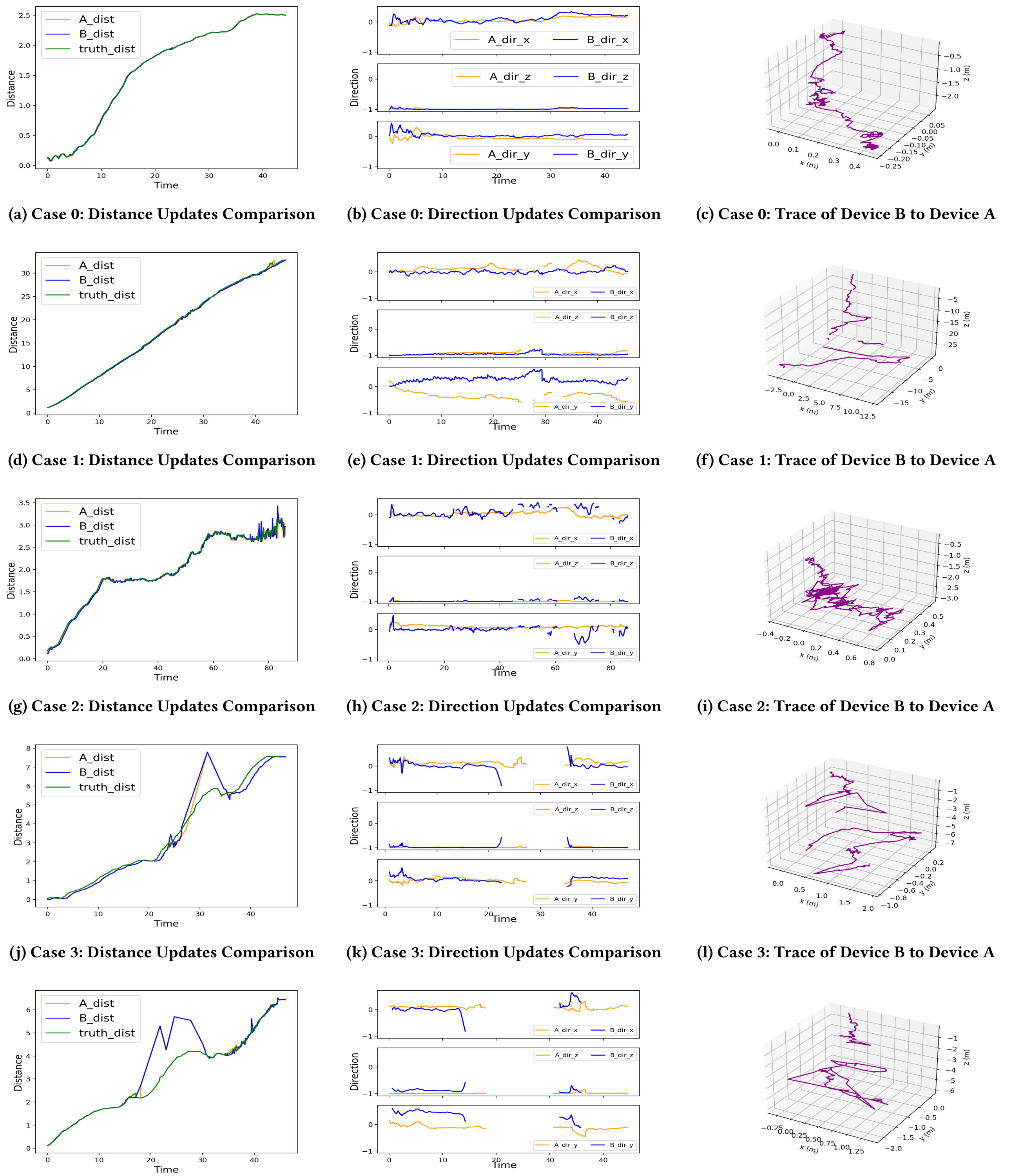
3.2 Measurements

The experiment results are presented in Figure 3, where the relative distance and time are measured in meters and seconds, respectively.
Ideal Case: Figure 3(a) to Figure 3(c) demonstrate the case without violating any aforementioned requirements. The Figure 3(c) shows the 3D trajectory of Device B from the perspective of Device A, while the other two figures show how the relative distance and direction of peer devices are reported.

It can be observed no points are missing, representing the position reports are always available for the entire session. Moreover, in Figure 3(a), the reported distances (lines in orange and blue in the figure) are the same to the true distance (green line in the figure).

With regards to the relative direction, the reports in both x and z axes are almost identical, as we can observe in the top and middle figures in Figure 3(b). In contrast, the direction updates in y axis are almost opposite in 3(b), since the y value represents the vertical height difference of two devices from their own perspective [10]. Because both devices are at the similar height in the experiment, however, the shown opposite property is subtle.
Range Impact: The impact of range is shown in Figure 3(d) to Figure 3(f). Reports for the relative distance between devices are always available and accurate, similar to the ideal case.

However, for the relative direction, absent reports about Device B begin to appear as the distance between two devices increases, for example, at about $t = 25s$, and the corresponding distance is 20 m, as shown in Figure 3(e). Also, the availability of reports are different to two devices: Device B is always able to track Device A while Device A not.
Takeaway: The position reports are always reliable, until the distance between devices exceeds $20m$.
Orientation Impact: In this case, we investigate how the screen orientation influences the devices' interaction. In this experiment, the screen of Device B is in the portrait orientation at first, then



(a) Case 0: Distance Updates Comparison

(b) Case 0: Direction Updates Comparison

(c) Case 0: Trace of Device B to Device A

(d) Case 1: Distance Updates Comparison

(e) Case 1: Direction Updates Comparison

(f) Case 1: Trace of Device B to Device A

(g) Case 2: Distance Updates Comparison

(h) Case 2: Direction Updates Comparison

(i) Case 2: Trace of Device B to Device A

(j) Case 3: Distance Updates Comparison

(k) Case 3: Direction Updates Comparison

(l) Case 3: Trace of Device B to Device A

(m) Case 4: Distance Updates Comparison

(n) Case 4: Direction Updates Comparison

(o) Case 4: Trace of Device B to Device A

Figure 3: Restrictions of Nearby Interaction in Practice

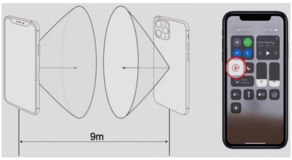


Figure 4: Nearby Interaction Requirements

Figure 5: CUWB Anchors (2 black and 2 white, attached to the walls

switched to the landscape at about $t = 24s$, and stays in the landscape mode until the end. The results are presented in the Figure 3(g) to Figure 3(i).

The reports for the relative distance are also reliable, no matter what the screen orientation of Device B is, as Figure 3(g) shows. For the relative direction, in Figure 3(h), we observe reports about Device A are not always available after Device B's screen is changed.
Takeaway: The relative direction is more vulnerable to the screen orientation impact than the relative distance. Once the screen is not in the portrait mode, the reports become intermittent.
Line of Sight Feature Impact: Figure 3(j) to Figure 3(l) demonstrate the influence of line of sight feature. At the first period from $t = 0s$ to $23s$, Device B is facing to Device A with its back camera. Then, at the second period from $t = 23s$ to $35s$, the holder of Device B turns around and becomes the solid obstacle in the line of sight between two devices. Lastly, at $t = 35s$, the holder of Device B turns around again so the line of sight is clear again until the end.

From the Figure 3(j) and Figure 3(k), it can be observed during the first and third period, the framework provides reliable reports. However, for the second period, the violation of line of sight feature shows its unique impact. *First*, there is a severe fluctuation in the reported relative distances. And the deviation from the truth means the updates, although available, are no longer accurate. *Second*, the direction updates to both devices are unavailable.
Takeaway: The line of sight feature influences both the relative distance the direction reports: the distance reports are not accurate while the direction updates are entirely unavailable.
Multiple Violations: In practice, the users movement may violate more than one requirement. Figure 3(m) to Figure 3(o) show the results of such a case.

At about $t = 15s$ to $32s$, the holder of Device B is the obstacle in the line of sight of two devices, similar to the previous case. Correspondingly, the distance reports are no longer accurate and direction reports are not available. During the following period, two devices resume to face each other with back camera and the line of sight is clear of obstacles. But the screen of Device B is changed to the landscape. In the Figure 3(n), it can be observed the direction updates of Device A are unavailable to Device B.
Takeaway: The result for the case involving multiple violations can be roughly seen as the combination of results of case with single corresponding violation.

4 ERROR ANALYSIS

The purpose of error analysis is to detect when errors happen and their potential reasons. Our analysis reveals, according to the magnitude of distance deviation and length of error duration, there are three types of errors. Their characteristics are listed as follows.

- Type I: transient error, reporting erroneous distance and missing relative direction. This type of error can be seen as a glitch and may happen in all cases. It is difficult to be recognized in Figure 3, because it is short-lived, with the interval of at most 100 ms between neighbouring reports. Also, its deviation from the truth is highly limited.
- Type II: persistent error, usually lasting longer than $1s$. It happens when participating devices violate the line of sight requirement, as shown from $t = 23s$ to $35s$ in Figure 3(j) and Figure 3(k).
- Type III: moderate error, is between the other two types, in terms of error duration and distance deviation magnitude. The direction update also remains missing. This type of error exists in cases involving violations.

Table 1: Error analysis results for the measurement study

Case	Device	Total Records	Total Time	Error Time	Type I	Type II	Type III
0	A	2556	44	0	0	0	0
	B	2578	44	0	1	0	0
1	A	2258	46	5.5	24	0	54
	B	2629	46	0	45	0	1
2	A	4840	86	0	16	0	0
	B	3695	86	22.4	5	0	154
3	A	2191	47	7.4	2	2	0
	B	1942	47	12.6	0	2	21
4	A	1853	46	12.5	10	9	1
	B	1155	46	27.6	4	7	102

Table 1 presents the quantitative results of error analysis for all five cases. For each row, the total number of records for both devices is reported in the third column. The next two columns report the duration of each experiment and accumulated length of error time in each experiment in seconds. The number of errors for each type is reported in the last three columns.

It can be observed the transient error takes place in all cases, consistent with the nature of glitch. Next, the persistent error only occurs in last two cases where an obstacle (holder of Device B) is present in the line of sight between devices. Finally, the duration of accumulated error time accounts for from 12% (in Case 1 Device A) to 60% (in Case 4 Device B) in cases with violations, depending on the users' specific activity. The long-term reliability of position reports may lead to participants having difficulty in understanding each other's intention in the experience.

5 CONCLUSION

In this work, we for the first time explore the reliability of the peer tracking framework utilizing the UWB chip in MuAR. The measurement results reveal the framework may produce erroneous reports because of its restrictions on user activity range, screen orientation and the presence of physical obstacles. We plan to develop a solution to achieve the reliable peer tracking in MuAR.

ACKNOWLEDGMENTS

We appreciate the constructive comments from the reviewers. This work is supported in part by the NSF grant CNS-2007153 and gift funding from Adobe Research.

REFERENCES

[1] Apple. 2017. Arcamera. https://developer.apple.com/documentation/arkit/arcamera.

[2] Apple. 2017. Arconfiguration.worldalignment.gravity. https://developer.apple.com/documentation/arkit/arconfiguration/worldalignment/gravity.

[3] Apple. 2017. Arkit. https://developer.apple.com/documentation/arkit/.

[4] Apple. 2018. Creating a multiuser ar experience. https://developer.apple.com/documentation/arkit/creating_a_multiuser_ar_experience.

[5] Apple. 2020. Direction. https://developer.apple.com/documentation/nearbyinteraction/ninearbyobject/3601161-direction.

[6] Apple. 2020. Distance. https://developer.apple.com/documentation/nearbyinteraction/ninearbyobject/3601163-distance.

[7] Apple. 2020. Initiating and maintaining a session. https://developer.apple.com/documentation/nearbyinteraction/initiating_and_maintaining_a_session.

[8] Apple. 2019. Iphone 11 pro and iphone 11 pro max: the most powerful and advanced smartphones. https://www.apple.com/newsroom/2019/09/iphone-11-pro-and-iphone-11-pro-max-the-most-powerful-and-advanced-smartphones/.

[9] Apple. 2018. Multipeerconnectivity. https://developer.apple.com/documentation/multipeerconnectivity.

[10] Apple. 2020. Nearby interaction. https://developer.apple.com/documentation/nearbyinteraction.

[11] Apple. 2018. Swiftshot. https://developer.apple.com/documentation/arkit/swiftshot_creating_a_game_for_augmented_reality.

[12] Apple. 2020. Ultra wideband availability. https://support.apple.com/en-us/HT212274/.

[13] Ciholas. 2020. Ciholas data protocol (cdp). https://pypi.org/project/cdp-py/.

[14] Ciholas. 2020. Cuwb bernoulli system. https://cuwb.io/docs/v3.0/installation-and-configuration/quick_start_guide/.

[15] Eg Su Goh, Mohd Shahrizal Sunar, and Ajune Wanis Ismail. 2019. 3d object manipulation techniques in handheld mobile augmented reality interface: a review. *IEEE Access*, 7, 40581–40601.

[16] Google. 2018. Arcore. https://developers.google.com/ar.

[17] Google. 2018. Arcore camera. https://developers.google.com/ar/reference/java/com/google/ar/core/Camera.

[18] Google. 2018. Justaline. https://justaline.withgoogle.com.

[19] Juan Londoño. 2021. The increased adoption of augmented and virtual reality and its challenges: a primer. https://www.americanactionforum.org/insight/the-increased-adoption-of-augmented-and-virtual-reality-and-its-challenges-a-primer/.

[20] Kathryn MacCallum and David Parsons. 2019. Teacher perspectives on mobile augmented reality: the potential of metaverse for learning. In *World Conference on Mobile and Contextual Learning*, 21–28.

[21] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. 2015. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31, 5, 1147–1163.

[22] Oberlo. 2020. Ar apps. https://www.oberlo.com/blog/augmented-reality-apps.

[23] Ian Oppermann, Matti Hämäläinen, and Jari Iinatti. 2004. *UWB: theory and applications.* John Wiley & Sons.

[24] Reportlinker. 2021. Augmented reality & mixed reality market - growth, trends, covid-19 impact, and forecasts (2021 - 2026). https://www.reportlinker.com/p06129739/Augmented-Reality-Mixed-Reality-Market-Growth-Trends-COVID-19-Impact-and-Forecasts.html?utm_source=GNW.

[25] Bektur Ryskeldiev, Yoichi Ochiai, Michael Cohen, and Jens Herder. 2018. Distributed metaverse: creating decentralized blockchain-based model for peer-to-peer sharing of virtual spaces for mixed reality applications. In *Proceedings of the 9th Augmented Human International Conference*, 1–3.

[26] Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, and Mika Ylianttila. 2021. A survey on mobile augmented reality with 5g mobile edge computing: architectures, applications, and technical aspects. *IEEE Communications Surveys & Tutorials*, 23, 2, 1160–1192.

[27] Matthew Sparkes. 2021. What is a metaverse. (2021).