ADA: Augmented-Reality Development Adaptor for Supporting AR glass

Heeyoon Jeong
VR/AR Research Center
Korea Electronics Technology Institute
Seoul, South Korea
nooyix@keti.re.kr

Yeseul Son
VR/AR Research Center
Korea Electronics Technology Institute
Seoul, South Korea
yeseul0323@keti.re.kr

Yoonmo Yang
VR/AR Research Center
Korea Electronics Technology Institute
Seoul, South Korea
yym064@keti.re.kr

Kwang-Soon Choi
VR/AR Research Center
Korea Electronics Technology Institute
Seoul, South Korea
lenon@keti.re.kr

***Abstract*—AR technology and AR devices have rapidly evolved, and the AR market is also very active. However, content creators face challenges in adapting their creations to different devices, as each device requires a distinct development approach. Moreover, hardware manufacturers often encounter difficulties in providing sustained software support for content development.**

In this study, we define data profiles for four distinct AR devices and propose the ADA SDK within the Unity framework. This ADA SDK aims to facilitate content development for multiple AR devices, providing content creators with a more streamlined approach. We demonstrate the potential for a single content piece developed using the ADA SDK to be adaptable across various devices through simulated testing within a simple Unity application. Building upon this data and SDK, we anticipate the emergence of a platform that seamlessly integrates AR capabilities, catering to a diverse range of devices. Furthermore, we expect this research to contribute to increased adoption of AR devices by small and medium-sized manufacturing companies, ultimately fostering the activation of the AR market.

***Index Terms*—Augmented Reality, Mixed Reality, Authoring Tool, AR glass, AR SDK**

I. INTRODUCTION

In recent times, rapid advances in meta-verse technology have led to significant advances in both hardware and software. This year's shipment is estimated to be 780,000 units as various manufacturers have entered the market due to a surge in demand for Augmented Reality(AR) equipment. However, in this growth, several important tasks remain in the AR content development area.

Primarily, developers and content creators deal with failures due to the diversity of AR hardware devices. While entities like Oculus and HoloLens offer development SDKs [1], [2], but many of AR devices do not provide dedicated software and only support smartphone tethering. Various studies are underway supported non-experts and content development designers in overcoming the challenges of AR development. Studies are being conducted focusing on creating user-friendly AR content development tools and platforms [3]–[5].

Secondly, developing AR software or content requires familiarity with a variety of technologies and tools. The complexity of this process is an obstacle for those who are not familiar with the complexity of AR. However, there is still a limit to the infrastructure that supports AR development through platforms such as Unity [6], a popular AR development environment. The paucity of educational resources, comprehensive documentation, sample codes, and developer support hampers the ability of AR content creators to fully harness Unity's potential.

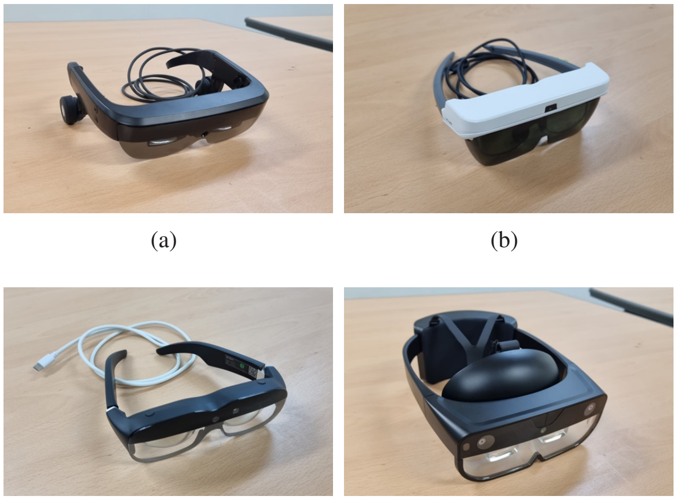
To overcome these challenges, it is essential to address two key areas. First, a robust test environment must be established to improve accessibility and affordability of AR devices while facilitating developer experiments. Second, cooperative efforts are needed to strengthen AR development using Unity. This includes developing a community of developers to create an environment that will help them take advantage of their comprehensive education, resource provision and Unity capabilities.

The focal point of this study is to present a holistic solution. We propose the creation of an integrated SDK structure and guide tailored for AR content creation across diverse devices. This solution is expected to promote AR device development and foster an ecosystem that provides a more convenient environment for developers and content creators together.

Subsequent sections provide complex details, including the structure of the AR spectacles sensory apparatus and Augmented-reality Development Adapter(ADA) system. In addition, we aim to demonstrate the feasibility of this approach by using ADA to create demo applications and validate their potential as effective SDK.

II. AR GLASS FEATURE

AR technology encompasses a wide array of advanced features such as real-time tracking, environmental sensing, 3D registration, and intelligent user interaction. Among the



(a) (b) (c) (d)

Fig. 1: (a): GTM200 Plus - Panacea, (b): Smart AR Glass - LatinAR, (c): XR-90FC - Elvision, (d): A21M XR Glass - PnC Solution

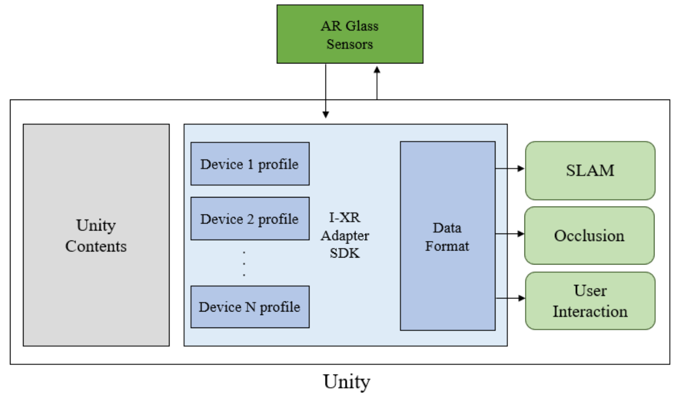


Fig. 2: ADA(Augmented-Reality Development Adaptor) Architecture. Convert to the data format specified by the ADA profile. It implements functions such as slam, occlusion, and interaction that can create AR content with converted standardized data.

prominent AR head-mounted displays (HMDs) in the current market, the Hololens2 [7] stands out for its integration of cutting-edge sensors. This device includes cameras for precise head tracking, infrared (IR) cameras that facilitates accurate eye tracking, a time-of-flight (TOF) depth sensor for spatial awareness, an inertial measurement unit (IMU) sensor for orientation tracking, and an RGB camera for capturing real-world visuals. Similarly, the Meta Quest Pro [8], renowned for its See-through AR capabilities, includes a depth sensor for depth perception, an RGB camera for capturing the environment, an ambient light sensor for adjusting display settings based on lighting conditions, and eye-tracking cameras for gauging user attention and focus. While various devices incorporate IMU sensors and cameras, the consensus is that the inclusion of 6-degree-of-freedom (6-DOF) data for user tracking alongside camera image data is paramount for the implementation of even basic AR functionalities, as widely cited in references encompassing different AR devices.

A. Sensor Data Format

The ADA SDK supports the four device profiles illustrated in Fig 1. These devices are categorized as (a), (b), (c), and (d), each offering distinct features based on their characteristics. Devices (a), (b), and (c) operate in a tethered mode, necessitating integration with an Android smartphone. In contrast, device (d) features an independent Android OS, enabling autonomous operation without the need for smartphone integration.

Each device profile is developed within the ADA SDK, facilitating the provision of features and data processing methods tailored to their respective attributes. Furthermore, the data collected from diverse devices is transformed into a standardized data format, enabling the SDK to process data uniformly across different devices.

These profiles and data conversion mechanisms enhance the ADA SDK's ability to support a wide range of devices, providing an adaptable environment to address the diversity of AR devices. This, in turn, is expected to facilitate more convenient development and content creation considering the diversity of AR devices.

1) IMU sensor: Inertial Measurement Unit (IMU): An IMU consists of an accelerometer, a gyroscope and a magnetic system sensor. In AR Glass, IMU is used to track user head motion. Head motion tracking detects changes in the user's perspective and maintains the visual consistency and sense of reality of virtual objects.

2) RGBD Image: RGBD Image: The image sensor captures video streams from the camera and provides video data that can be displayed on the AR Glass display. RGBD image sensors transmit real-time video streams to AR Glass systems and are used to implement various functions such as environmental recognition, object tracking, and virtual object placement.

III. System Development

A. Architecture

AR systems are being used and developed in various fields such as education and industry [9]–[15]. Currently, only IMU and camera sensors are available as support equipment. The ADA defines the sensor and data format of each device as a profile, as shown in Figure 2. Then, print it in a standardized format.

When creating content through ADA, only rotational values using IMU sensors can be applied for the four types of devices shown in Fig 1. Data from these devices is transmitted to a Unity app via serial communication. Subsequently, the data is formatted into standardized formats according to the profiles defined for each device.

In the future, by incorporating spatial recognition algorithms based on input images and IMU sensors [16], [17], it will

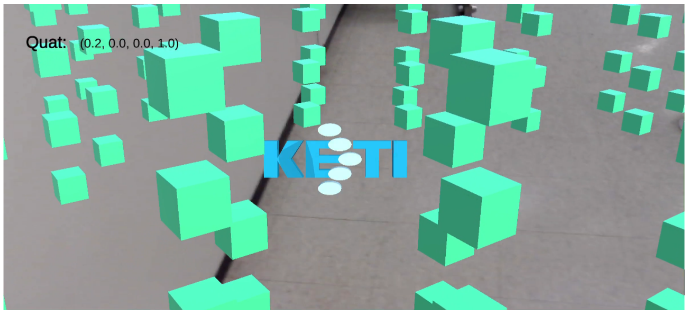


Fig. 3: An AR application developed using the ADA SDK showcases static virtual boxes anchored within the environment, while a logo dynamically follows the user's movements. This application is designed to operate seamlessly across all AR glasses supported by ADA.

be possible to achieve a more seamless AR experience. Furthermore, if ADA were to support various sensors attached to the devices, such as infrared sensors, light sensors, and distance measurement sensors, a wide range of interactions could be implemented, resulting in a richer user experience. Also, ADA will also incorporate features such as SLAM, occlusion rendering, and enhanced user interface capabilities.

B. Demonstration of ADA

To validate the capability of supporting multiple AR devices with a unified content approach, a simple Unity app was developed using the ADA SDK. Through this demonstration, the effectiveness of an SDK that enables AR development across various manufacturers' devices is substantiated.

Within the app, stationary objects and objects that follow the user's movements are placed. Leveraging IMU data and RGBD image data from each device, virtual object placement and virtual camera movements are controlled. The content developed through the ADA SDK showcases its immediate applicability across multiple devices, tailored to Unity settings. This content app is compatible with all four types of AR devices in Fig 1, proving that ADA SDK can support AR functions for multiple devices, and the content screen configuration can be seen on Fig 3.

However, the current system faces certain limitations due to device interoperability and interaction challenges. Firstly, while the system supports augmenting virtual objects based on location data, it lacks direct interaction between users and objects. This limits users from experiencing more immersive and natural interactions with the content. Secondly, the absence of hand gesture inputs confines users to limited interaction through device UIs, potentially diminishing user engagement and experience.

These issues can be addressed and improved upon through subsequent development and expansion of AR functionalities. Moreover, with the inclusion of such capabilities, the ADA SDK could evolve to fulfill a broader role as an AR SDK, enhancing its overall utility.

IV. Conclusion

With As AR technology continues to evolve, devices from various manufacturers are appearing one after another. To effectively accommodate this diversity and ensure seamless content creation and interaction across different devices, the development of a versatile AR SDK is crucial. These SDKs facilitate content creation and interaction between devices from multiple manufacturers. The current system's demonstration validates the feasibility of integrating devices from various manufacturers using an SDK. However, further extended experiments and tests involving a wider range of devices and scenarios are needed to establish the practicality and reliability of SDK applications.

In this paper, we identified the shortcomings of existing systems and proposed solutions. It is expected that the development of Universal AR SDK, which incorporates gesture recognition for direct interaction and integrates various devices, will improve both AR content production and user experience. Furthermore, validating the practicality of SDK experimentally integrating devices from multiple manufacturers highlights its potential contribution to advances in AR technology.

To address the inconveniences faced by AR device developers and content creators, several suggestions are put forth. This paper introduces SDK system ADA to support AR development at Unity, integrates various AR functions into the Unity engine, and provides developers with the resources and sample code they need. These efforts are aimed at revitalizing the AR device development ecosystem, simplifying AR content development, and fostering an efficient environment for developers and content creators.

In conclusion, this paper argues that ADA, which promotes Unity integration, relieves inconvenience for AR device developers and content creators and focuses on developer education and community participation. This approach is expected to not only increase the utilization of AR devices but also greatly contribute to the expansion of the AR market.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00962, Development of industrial XR framework for smart manufacturing)

References

[1] Meta. Oculus developer center. [Online]. Available: https://developer.oculus.com/

[2] Microsoft. Mixed reality. [Online]. Available: https://learn.microsoft.com/en-us/windows/mixed-reality/

[3] B. MacIntyre, M. Gandy, J. Bolter, S. Dow, and B. Hannigan, "Dart: The designer's augmented reality toolkit," in *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.* IEEE, 2003, pp. 329–330.

[4] A. Hampshire, H. Seichter, R. Grasset, and M. Billinghurst, "Augmented reality authoring: generic context from programmer to designer," in *Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, 2006, pp. 409–412.

[5] N. Ashtari, A. Bunt, J. McGrenere, M. Nebeling, and P. K. Chilana, "Creating augmented and virtual reality applications: Current practices, challenges, and opportunities," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–13.
[6] Unity. Unity. [Online]. Available: https://unity.com/
[7] Microsoft. Hololens2. [Online]. Available: https://www.microsoft.com/ko-kr/hololens/hardware
[8] Meta. Metaquestpro. [Online]. Available: https://www.meta.com/kr/en/quest/quest-pro/
[9] L. F. de Souza Cardoso, F. C. M. Q. Mariano, and E. R. Zorzal, "A survey of industrial augmented reality," *Computers & Industrial Engineering*, vol. 139, p. 106159, 2020.
[10] M. Rettinger and G. Rigoll, "Defuse the training of risky tasks: Collaborative training in xr," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 695–701.
[11] V. Lehrbaum, A. MacWilliams, J. Newman, N. Sudharsan, S. Bien, K. Karas, C. Eghtebas, S. Weber, and G. Klinker, "Enabling customizable workflows for industrial ar applications," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 622–630.
[12] Y. Cao, T. Wang, X. Qian, P. S. Rao, M. Wadhawan, K. Huo, and K. Ramani, "Ghostar: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 521–534.
[13] G. Huang, X. Qian, T. Wang, F. Patel, M. Sreeram, Y. Cao, K. Ramani, and A. J. Quinn, "Adaptutar: An adaptive tutoring system for machine tasks in augmented reality," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–15.
[14] A. R. Fender and C. Holz, "Causality-preserving asynchronous reality," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–15.
[15] A. Guo, I. Canberk, H. Murphy, A. Monroy-Hernández, and R. Vaish, "Blocks: Collaborative and persistent augmented reality experiences," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–24, 2019.
[16] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of imu and vision for absolute scale estimation in monocular slam," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 287–299, 2011.
[17] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.