2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)

2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct) | 979-8-3503-2891-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISMAR-Adjunct60411.2023.00023

Reality Stack I/O: A Versatile and Modular Framework for Simplifying and Unifying XR Applications and Research

Florian Kern*
University of Würzburg
Würzburg, Germany

Marc Erich Latoschik†
University of Würzburg
Würzburg, Germany

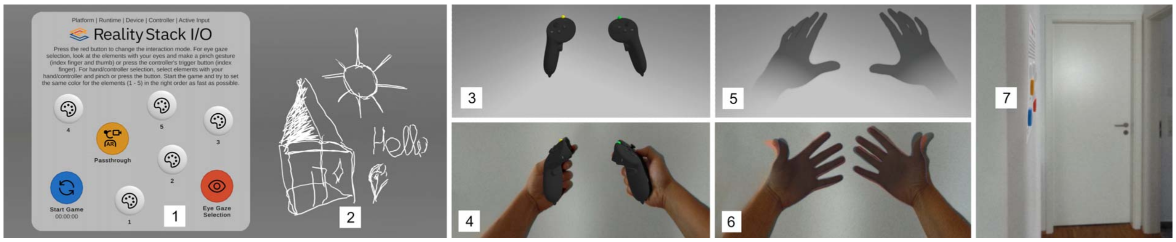

Figure 1: The Reality Stack I/O example, implemented with Unity, consists of a 2D user interface (1), a 3D sketching tool (2), and a virtual-to-physical surface alignment technique (3ViSuAl [4]). In this scenario, the user operates a Meta Quest Pro device, interacting via controllers (3, 4), hand tracking (5, 6), and eye tracking. Pressing the orange passthrough UI button switches between VR (3, 5) and VST AR (4, 6). The 3D sketching tool enables basic drawings (2), and 3ViSuAl makes it easy to align virtual surfaces with flat physical surfaces. Here, the user has aligned the 2D user interface with a wall, allowing for direct touch interaction (7).

Abstract

This paper introduces Reality Stack I/O (RSIO), a versatile and modular framework designed to facilitate the development of extended reality (XR) applications. Researchers and developers often spend a significant amount of time enabling cross-device and cross-platform compatibility, leading to delays and increased complexity. RSIO provides the essential features to simplify and unify the development of XR applications. It enhances cross-device and cross-platform compatibility, expedites integration, and allows developers to focus more on building XR experiences rather than device integration. We offer a public Unity reference implementation with examples.

Index Terms: Human-centered computing—Virtual reality—; Human-centered computing—Mixed / augmented reality—;

1 Introduction

Spatial computing is an essential step towards seamlessly integrating physical and virtual environments for extended reality (XR) workspaces. It leverages technologies such as virtual reality (VR), augmented reality (AR), and mixed reality (MR) that enable users to interact with digital and real content in an intuitive, spatial way.

In particular, cross-reality (CR), also called transitional interfaces (TI), supports simultaneous use and transition between different points on the reality-virtuality continuum [7,9]. CR/TI are promising approaches for understanding and exploring spatial data and related information, that need to be explored in HCI research [1, 3, 8].

For example, Schröder et al. [8] presented analytical lenses for understanding dyadic collaboration in transitional interfaces, Jetter et al. [2] explored VR as a design tool for sketching and simulating spatially-aware interactive spaces, and Kern et al. [5] investigated controller-based virtual tap- and swipe keyboards in VR and video see-through (VST) AR.

While the design of these applications already requires extensive knowledge, another major challenge becomes apparent during development and research: the variety of XR devices and platforms.

XR devices offer a wide range of capabilities that are continually expanded. For example, early devices (e.g., Oculus Rift S) only support head and controller tracking. In comparison, recent devices (e.g., Meta Quest Pro or Pico 4 Enterprise) also provide sensors for hand-, eye-, and face tracking, as well as MR. Therefore, researchers and developers often spend a significant amount of time enabling cross-device and cross-platform compatibility rather than building XR experiences for visualization, interaction, design, or collaboration.

For Unity, an established cross-platform game engine for XR development, frameworks such as Mixed Reality Toolkit (MRTK)[1] or XR Interaction Toolkit (XRI)[2] have emerged as popular solutions for developing applications across realities, devices, and platforms. Focused on the OpenXR standard, these frameworks provide flexible input systems and basic components for MR interactions and interfaces. However, XR device manufacturers often make their latest features available only through their native XR plugins, rather than directly integrating the OpenXR standard by default. As a result, only a limited number of devices are supported, and incorporating new features can be highly time-consuming or even impossible.

In contrast, frameworks like the Virtual Reality Toolkit (VRTK)[3] and UltimateXR[4] simplify development by using native XR plugins instead of relying solely on the OpenXR standard. While these frameworks support a wide range of consumer XR devices, they focus primarily on interaction features for VR rather than CR/TI. Consequently, they offer limited support for technological advances such as CR and MR, eye- and face tracking, or spatial anchors.

Another well-established virtual reality platform is SteamVR[5], best known from the digital distribution platform Steam. SteamVR serves as an interface between XR hardware and software and is renowned for its high cross-device compatibility. However, the integration for cross-platform XR is restricted. Concretely, it does not offer native support for Android-based standalone XR devices, which limits its reach in the increasingly diverse XR ecosystem.

*e-mail: florian.kern@uni-wuerzburg.de
†e-mail: marc.latoschik@uni-wuerzburg.de

[1] https://learn.microsoft.com/windows/mixed-reality/mrtk-unity/
[2] https://docs.unity3d.com/Manual/com.unity.xr.interaction.toolkit/
[3] https://www.vrtk.io/
[4] https://www.ultimatexr.io/
[5] https://store.steampowered.com/app/250820/SteamVR/

DOI 10.1109/ISMAR-Adjunct60411.2023.00023

With Reality Stack I/O (RSIO), we aim to bridge the gap between feature-rich interaction frameworks and the most advanced XR device features. RSIO builds on the OpenXR standard and, if the required functionality is not yet available, on manufacturers' native XR plugins. It ensures flexibility and versatility, making it compatible with Microsoft Windows and Android-based XR devices. Our approach enhances cross-device and cross-platform compatibility, expedites integration, and allows developers to focus on building XR experiences rather than device integration. From a research perspective, RSIO can support the replicability of previous experiments and facilitate direct comparisons between XR devices.

Reality Stack is our vision to provide reusable and publicly available software components for XR applications and research. We offer a public Unity reference implementation[8] with examples.

2 Design Requirements

In developing RSIO, our primary goal was to expedite and simplify XR development and research. Therefore, we defined the following design requirements: The RSIO architecture should be inherently flexible to allow easy and timely extensions. RSIO should be compatible with various XR devices, established interaction frameworks, and custom solutions for social XR, sketching, and text input. In addition, it should be able to function as a standalone framework. RSIO should be user-friendly, and provide examples and documentation.

3 Reference Implementation

For our reference implementation, we use Unity 2021.3 LTS. Unity is a widely used and versatile game engine that provides extensive support for XR development, making it an ideal platform for our purposes. Using a long-term support (LTS) version ensures stability and consistency. Fig. 2 shows the architecture of RSIO.

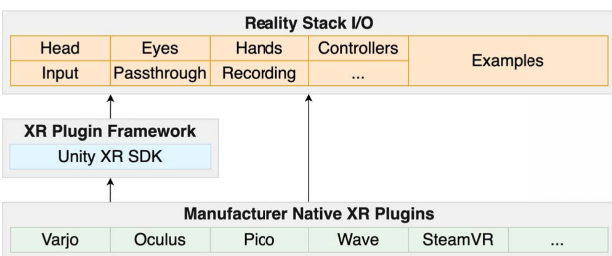

Figure 2: The architecture of Reality Stack I/O. We use Unity XR SDKs and, when features are not yet supported, the manufacturers' native XR plugins.

RSIO provides developers with access to lightweight interfaces that abstract the OpenXR standard and, if features are not yet supported, utilize manufacturers' native XR plugins. We provide essential features, including head tracking, eye tracking, hand tracking, controller tracking, hand gestures, and input capabilities. Considering the growing interest in and accessibility of MR devices, we also include passthrough support. The system also provides developers with features for aggregated, unified data access. For example, the generic input feature unifies user input from controller buttons, hand gestures, and traditional keyboard and mouse.

XR devices usually offer similar tracking sources with a head-mounted display (HMD) and hand controllers. However, controllers can greatly vary in shape and grasping possibilities [4]. Therefore, our system provides tracking anchors that are, by default, attached to fingertips and controller models. The concept of tracking anchors derives from previous research [4, 6], which proposed uniform reference points on XR controllers (i.e., a stylus tip anchor) for 2D interactions (in 3D), handwriting, and sketching.

Fine-grained recording of raw (i.e., positions/rotations) and unified (i.e., generic input) data is also essential for XR research. We offer a flexible CSV-based data recording interface that empowers researchers to conduct in-depth analyses of their user studies and develop replay capabilities for recorded user sessions. Each feature (e.g., eye or hand tracking) can implement the recording interface and is responsible for providing its CSV header and data.

Showcasing our Unity reference implementation, we provide the Oculus plugin designed for Meta/Oculus devices, as depicted in Fig. 3. The plugin supports eye-, controller-, and hand tracking, recognizes hand gestures, supports passthrough, provides tracking anchors, and implements the data recording interface. In addition, it incorporates a generic input feature that merges various input modalities like hand gestures and controller-, keyboard- and mouse buttons. Our plugin relies on the official OVRCameraRig and can be easily extended with recent Meta features like face- and body-tracking.

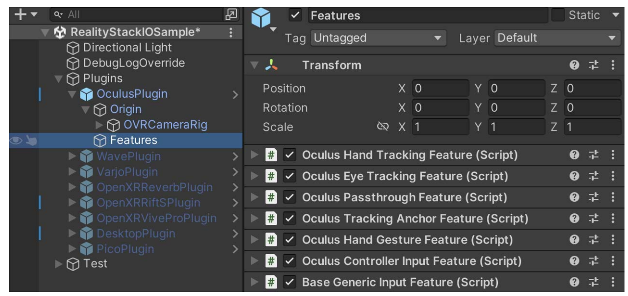

Figure 3: The RSIO Oculus plugin is designed for Meta/Oculus devices, supports various features, and is easily extendable.

4 Examples

We support researchers and developers by offering various examples of how to use RSIO. In addition to XR device and platform integration, we provide examples for spatial interaction and spatial input. Examples include distance-based UI interaction with controllers, hand gestures, and eye-gaze pointing approaches, as well as direct touch-based solutions (See Fig. 1). We also integrate RSIO with a public social XR platform, Ubiq [10], and the Off-The-Shelf Stylus framework [4] for 2D interactions, sketching, and handwriting.

5 Limitations and Future Work

While RSIO aims to simplify XR development, every new framework has its learning curve. To assess usability, we plan to conduct a user evaluation involving novice and experienced XR developers. Since RSIO introduces an additional layer, there might be performance implications. A technical evaluation can identify optimization opportunities. In a rapidly evolving XR field, RSIO faces development, support, and maintenance challenges, but we try to integrate XR devices and platforms continually. While other interaction frameworks primarily target consumer XR devices, we also plan to include motion-tracking systems like OptiTrack[6] and TheCaptury[7].

6 Conclusion

Reality Stack I/O is a versatile and modular solution for developing XR applications and research. Our approach enhances cross-device and cross-platform compatibility, expedites integration, and allows developers to focus more on building XR experiences rather than device integration. We offer a publicly available Unity reference implementation[8] accompanied by various examples.

[6]https://optitrack.com/
[7]https://captury.com/
[8]https://go.uniwue.de/realitystack-io

ACKNOWLEDGMENTS

Our research has been funded by the German Federal Ministry of Education and Research (BMBF) in the project ViLeArn More (Reference: 16DHB2214) and by the Bavarian State Ministry For Digital Affairs in the project XR Hub (Reference: A5-3822-2-16). We thank Florian Heinrich for manuscript discussions and Peter Kullmann for architectural discussions. We are also thankful to the anonymous reviewers for their helpful comments.

REFERENCES

[1] R. Grasset, J. Looser, and M. Billinghurst. Transitional interface: concept, issues and framework. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 231–232. IEEE, Santa Barbara, CA, USA, Oct. 2006. doi: 10.1109/ISMAR.2006.297819

[2] H.-C. Jetter, R. Rädle, T. Feuchtner, C. Anthes, J. Friedl, and C. N. Klokmose. "In VR, everything is possible!": Sketching and Simulating Spatially-Aware Interactive Spaces in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–16. ACM, Honolulu, HI, USA, Apr. 2020. doi: 10.1145/3313831.3376652

[3] H.-C. Jetter, J.-H. Schröder, J. Gugenheimer, M. Billinghurst, C. Anthes, M. Khamis, and T. Feuchtner. Transitional Interfaces in Mixed and Cross-Reality: A new frontier? In *Interactive Surfaces and Spaces*, pp. 46–49. ACM, Lodz Poland, Nov. 2021. doi: 10.1145/3447932.3487940

[4] F. Kern, P. Kullmann, E. Ganal, K. Korwisi, R. Stingl, F. Niebling, and M. E. Latoschik. Off-The-Shelf Stylus: Using XR Devices for Handwriting and Sketching on Physically Aligned Virtual Surfaces. *Frontiers in Virtual Reality*, 2:684498, June 2021. doi: 10.3389/frvir.2021.684498

[5] F. Kern, F. Niebling, and M. E. Latoschik. Text Input for Non-Stationary XR Workspaces: Investigating Tap and Word-Gesture Keyboards in Virtual and Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2658–2669, May 2023. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi: 10.1109/TVCG.2023.3247098

[6] F. Kern, M. Popp, P. Kullmann, E. Ganal, and M. E. Latoschik. 3D Printing an Accessory Dock for XR Controllers and Its Exemplary Use as XR Stylus. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, VRST '21. Association for Computing Machinery, New York, NY, USA, 2021. event-place: Osaka, Japan. doi: 10.1145/3489849.3489949

[7] P. Milgram and F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information and Systems*, 77:1321–1329, 1994.

[8] J.-H. Schröder, D. Schacht, N. Peper, A. M. Hamurculu, and H.-C. Jetter. Collaborating Across Realities: Analytical Lenses for Understanding Dyadic Collaboration in Transitional Interfaces. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–16. ACM, Hamburg Germany, Apr. 2023. doi: 10.1145/3544548.3580879

[9] R. Skarbez, M. Smith, and M. C. Whitton. Revisiting Milgram and Kishino's Reality-Virtuality Continuum. *Frontiers in Virtual Reality*, 2:647997, Mar. 2021. doi: 10.3389/frvir.2021.647997

[10] A. Steed, L. Izzouzi, K. Brandstätter, S. Friston, B. Congdon, O. Olkkonen, D. Giunchi, N. Numan, and D. Swapp. Ubiq-exp: A toolkit to build and run remote and distributed mixed reality experiments. *Frontiers in Virtual Reality*, 3:912078, Oct. 2022. doi: 10.3389/frvir.2022.912078