Overcoming Challenges of Integrating Heterogeneous Commercial and Open-Source Tools in Extended Reality Applications

Kexin Wang, Eliot Winer
Iowa State University
Ames, Iowa
kexinw@iastate.edu, ewiner@iastate.edu

ABSTRACT

Extended Reality (XR), an umbrella term for Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), continues to be used in a wide variety of fields, including training, education, entertainment, and especially the military. For example, the Army requested $427 million in 2024 for the Synthetic Training Environment (STE) program, which features XR for many of the desired applications. For complex and immersive XR environments, interaction modalities are often required, including hand, eye, and body tracking, custom physical interactions, and high visual fidelity. Industry, academic researchers, and the open-source community have started to introduce equipment and software tools that address these needs. However, achieving these capabilities in a seamless integrated simulation environment often requires hardware and software from different vendors or open-source platforms. Ensuring compatibility between heterogeneous equipment and software in an XR simulation presents significant challenges for developers.

This paper presents two case studies that demonstrate the challenges of integrating heterogeneous commercial and open-source equipment and software into a single XR simulation. The development process used open-source software including OpenXR and a local laptop in a closed WiFi network to overcome the hardware and software issues. The first case is a portable MR welder training simulator that supports interaction with a customized welder frame interface and gun. This application requires integrating an Arduino, Vive Ultimate Tracker, and standalone XR headsets. The second case is a training application for veterinary euthanasia using captive bolt guns. This application requires the use of HMDs with Vive Ultimate Tracker to enable interaction with a physical bolt gun in an XR simulation across multiple headsets. A technical evaluation of both systems will be discussed, measuring several performance metrics including frame rate and latency. These use cases highlight the potential for developing complex XR environments using heterogeneous commercial and open-source equipment and software.

ABOUT THE AUTHORS

Kexin Wang is a Ph.D. student in Human-Computer Interaction and Computer Engineering at Iowa State University's Virtual Reality Application Center. She is working on developing general Extended Reality application for education and training purposes.

Eliot Winer is the director of the VRAC Research Center and professor of Mechanical Engineering, Electrical and Computer Engineering, and Aerospace Engineering at Iowa State University. Dr. Winer has over 25 years of experience working in extended reality and 3D computer graphics technologies on sponsored projects for the Department of Defense, Air Force Office of Scientific Research, Department of the Army, National Science Foundation, Department of Agriculture, Boeing, John Deere, and the Federal Highway Administration.

Overcoming Challenges of Integrating Heterogeneous Commercial and Open-Source Tools in Extended Reality Applications

Kexin Wang, Eliot Winer
Iowa State University
Ames, Iowa
kexinw@iastate.edu, ewiner@iastate.edu

INTRODUCTION

Extended Reality (XR) is an umbrella term for Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR) (Adriana Cárdenas-Robledo et al., 2022; Çöltekin et al., 2020; Kini et al., 2024). VR refers to a technology in which the user is fully immersed in a computer-generated environment. AR overlays computer-generated content onto the user's physical surrounding, allowing 3D digital elements to appear integrated with the real world. Mixed Reality (MR) integrates physical components into the virtual environment (VE). Users in MR are immersed in a computer-generated environment, but their interactions with physical objects can be synchronized, or trigger events, in the virtual scene. All three XR technologies are widely applied across domains such as technical communication (Heinonen, 2023), education (Jamah et al., 2022), military training (Garcia Estrada et al., 2024), and medical practice (Andrews et al., 2019). Research has shown that XR technologies offer significant benefits across different fields, especially in the military (Adriana Cárdenas-Robledo et al., 2022; Andrews et al., 2019; Chen & Konomi, 2022; Chuah, 2018; Çöltekin et al., 2020; Garcia Estrada et al., 2024; Heinonen, 2023; Jamah et al., 2022; Steven et al., 2023). Since the development of the Link Trainer in the 1930s U.S. military has a long history of using immersive simulation technologies for training purposes (Michael Moshell, 1993). Over the years, numerous projects have explored XR's role in military applications, including aviation training, combat environments, and decision making (Adam Vogel et al., 2018; Boyce et al., 2022; Caso, 2024; Garcia Estrada et al., 2024; Livingston et al., 2011; Soni & Kaur, 2023; Steven et al., 2023; Yoo & Ji, 2024).

With the growing demand for diverse XR content across various domains, different interaction modalities start to be integrated into XR applications to create unique experiences and achieve specific interactions. These modalities include eye tracking, body tracking, hand tracking, and even customized physical interfaces. For example, a flight simulation training application may incorporate a yoke, realistic control buttons, or specialized seating to enhance immersion (Zintl et al., 2024). A body tracking system can be used to track user movements in an XR application (Liang et al., 2022). Commercially developed interaction modalities often utilize proprietary hardware and software, which limits the interoperability and scalability of XR systems. Even when companies abide by open standards such as OpenXR, it can still be challenging to integrate different companies' hardware into a single XR application (HTC VIVE, 2025b; META QUEST, 2025b). For example, researchers have used the Vive Ultimate Tracker and Vive Focus 3 to enable body tracking in XR for teleoperation of humanoid robots (Bertrand et al., 2024). However, the first-generation Vive Tracker relies on external base stations (Lighthouses), making it hard to set up and use with growing portable HMDs (HTC VIVE, 2025a). The second-generation Vive Ultimate Tracker addresses this by using inside-out depth cameras, but it still depends heavily on Vive's proprietary head-mounted display (HMD) ecosystem (HTC VIVE, 2025a). Open-source hardware such as Arduino (Arduino, 2025) also plays a significant role in enabling customized interactions. Developers can integrate sensors, such as posture or vibration sensations, with Arduino boards and transmit the data to XR headsets like the HoloLens (Min et al., 2024). However, open-source hardware can be difficult to maintain, often lacks comprehensive documentation, and may require significant customization before integrating into XR applications. Hand tracking is a standard feature in many current HMDs and also available through third-party solutions like Leap Motion (Ultraleap, 2025). However, hand tracking accuracy and rendering capability on different HMDs can vary across devices (Reimer et al., 2023; Theodoridou et al., 2023; Weichert et al., 2013).

Depending on the fidelity and interaction requirements of a given XR application, it is often necessary to integrate modalities by different manufacturers, or research groups, and deploy them to different HMDs. This type of multi-source integration is referred to as using "heterogeneous" tools. In contrast, XR systems that only use tools from a single producer are referred to as "homogeneous" in this paper. Heterogeneous tools can offer developers broader

capabilities, support a wider range of interaction modalities, and serve as more cost-effective solutions. Heterogenous tools are especially needed in applications that require a high degree of realism, custom interactions, and compatibility across multiple HMDs. Using multiple heterogeneous tools in XR can offer significant benefits, but their implementation also presents notable challenges. For example, studies tend to focus on just one single modality (i.e., mode of interaction) (Sampige et al., 2024), rather than exploring how multiple modalities can be integrated effectively within one XR system. Second, when commercial tools are used, many researchers use homogeneous systems to avoid integration issues, but this approach can limit portability, fidelity, and accessibility for users with different HMDs (Caserman et al., 2019; Liang et al., 2022; Peer et al., 2018; Wang et al., 2021). For example, educational XR applications require portability to support deployment across various teaching locations with each having different HMDs (Qin et al., 2021). Current research on heterogeneous tools in XR systems mostly focuses on ensuring compatibility of a single XR application across different HMDs (An et al., 2024; Huang et al., 2024), rather than integrating multiple interaction modalities.

This paper explores the challenges and benefits of using heterogeneous XR systems through two use cases. The first use case is a welding simulation designed to provide a portable and realistic training experience. This MR application provides a welder interface with dials, switches and a physical welding gun that users interact with inside a VE. No existing commercial solutions provide the set of heterogenous tools needed for the desired interaction and portability in this MR application. The second use case is a veterinary training XR application intended for use in farm settings, outside traditional laboratory environments. Because the training emphasizes the handling and alignment of a euthanasia device, the application places greater importance on visual quality and hand-tracking fidelity. To expand the accessibility of the application, the system is designed to support different interaction methods, including hand tracking, controller tracking, and using a physical euthanasia device in VEs. For both cases, tracking physical objects is required. The Vive Ultimate Tracker is the most viable option due to its six degrees of freedom (6DoF) inside-out tracking capabilities and ease of mounting on real-world items. But the Vive Ultimate Tracker is typically used with HTC HMDs, and limited research has explored its integration with other HMD platforms. Other than physical tracking, each use case also introduced unique challenges, such as implementing customized interaction for the welder interface or ensuring accurate tracking to support precise alignment during the euthanasia training process.

BACKGROUND

Opportunity of Using XR technology

A literature review from 2022 summarized XR training benefits from 42 different papers published in recent years and identified four key benefits: motivation, performance, subjective satisfaction, and immersion (Kuhail et al., 2022). Another review also emphasized the use of XR in education, highlighting its potential to support distance learning, increase student engagement and motivation, and provide a repeatable training for high-risk or costly training scenarios (Jamah et al., 2022). XR has been applied in numerous military training applications as well including combat simulation, stress management, weapons training, and decision making (Boyce et al., 2022; Caso, 2024; Garcia Estrada et al., 2024; Gawlik-Kobylińska et al., 2020; Pallavicini et al., 2016; Soni & Kaur, 2023; Steven et al., 2023). In addition to education and training, XR improves communication due to its ability to provide real time 3D visualization (Heinonen, 2023; Yoo & Ji, 2024). Research by Yoo & Ji (2024) introduced a shared AR interface designed to provide users with visual cues and GPS mapping information during military operation. Their study compared participants' situational awareness and task performance with and without the AR interface. The findings indicated that the AR interface improved situational awareness but also increased cognitive workload and reduced effectiveness when the AR perspective was misaligned with the user's real-world orientation. XR applications can also support interactions beyond 3D visualization and serve as assistants during medical procedures and military operations (Andrews et al., 2019; Livingston et al., 2011). For example, research has integrated image processing with AR to identify key objects during military missions such as highlight hidden or critical elements and generate interactive maps for route planning and sharing (Livingston et al., 2011). However, these systems often face compatibility limitations, particularly regarding the types of HMDs or mobile devices they can support (Livingston et al., 2011). Similar challenges exist in medical XR applications, including difficulties in integrating sensors and the need for expensive or customized hardware to deliver mechanical feedback (Andrews et al., 2019).

Opportunities and Challenges of Adding Interaction Modalities into XR applications

Traditional XR systems typically include a display device such as a smartphone or HMD like the Oculus Quest (META QUEST, 2025a) or Vive Focus (HTC VIVE, 2025a). In the case of HMDs, it often comes with basic controllers like the Meta Quest Touch Pro (META QUEST, 2025a) or VIVE XR/Focus Series Controllers (HTC VIVE, 2025a). With the growing use of XR technology, interaction modalities such as eye tracking, hand tracking, and other customized hardware have begun to be integrated into XR applications (Buckingham, 2021; Min et al., 2024; Sampige et al., 2024). XR-SANS is a multi-modal framework that incorporates XR and eye tracking to analyze visual changes, specifically aiming to enhance the evaluation of astronaut visual function and contribute to a better understanding of Spaceflight-Associated Neuro-ocular Syndrome (SANS) (Sampige et al., 2024). Additional tracking devices can expand the range of possible interactions in XR applications, enabling capabilities such as full-body tracking, physical object tracking, and alignment between physical and virtual spaces.(Caserman et al., 2019; Hepke et al., 2024; Liang et al., 2022; Peer et al., 2018; Rozumnyi et al., 2025), The Vive Tracker is one of the most commonly used commercial tools for body tracking in XR research (Caserman et al., 2019; Liang et al., 2022; Peer et al., 2018; Wang et al., 2021). However, most applications using the Vive Tracker are developed exclusively for Vive HMDs, limiting cross-platform availability (Caserman et al., 2019; Liang et al., 2022). Researchers have also explored alternative approaches, such as using HMDs and controllers to generate three-point poses, combined with a depth-sensing camera sensor for body motion tracking (Rozumnyi et al., 2025). However, this method depends on estimation and requires users to remain within the sensor's limited field of view in order to generate accurate point cloud data. Enabling users to interact with VEs using their hands rather than controllers can enhance immersion, presence, and the overall effectiveness of interaction (Buckingham, 2021). Gesture recognition further improves the user experience by allowing intuitive input methods. Though literature has shown promising opportunities for hand tracking, it also has several challenges. Tracking errors caused by hardware limitations can also affect the usability in XR application. Modern HMDs, such as the Oculus Quest 3, natively include hand tracking, and commercial tools like Leap Motion can also offer additional capabilities, but performance can vary across platforms and devices (Reimer et al., 2023). Customized interaction methods can also enhance XR applications by incorporating specialized sensors or feedback mechanisms (Min et al., 2024; Qiu et al., 2024). For example, XR CUBE is a system designed to support multi-sensor integration and custom sensations within XR applications (Min et al., 2024). XR CUBE demonstrated the potential of connecting Arduino-based sensors to the Microsoft HoloLens AR HMD, enabling customized user inputs and feedback such as posture tracking for arm movement and vibration-based haptic feedback.

In complex system designs, multiple modalities are often wanted. However, integrating different interaction methods across different HMDs remains a significant challenge. Several studies and open-source solutions have tried to address this issue (An et al., 2024; Huang et al., 2024; Kern & Latoschik, 2023; OpenXR, 2025). For example, OpenXR provides a standardized approach aimed at supporting cross-HMD compatibility (OpenXR, 2025). However, OpenXR mainly focuses on HMD visual compatibility, and it is limited when it comes to adding additional hardware such as Arduinos, sensors, or trackers in a cross-platform XR application.

METHODOLOGY

The following section presents two case studies where heterogeneous tools were developed for an XR system. In both cases, the Vive Ultimate Tracker was used to track physical objects within the VE. Two potential system architectures were considered. The first involved using an HTC-dominated setup, where the Vive Tracker could connect directly to HTC standalone HMDs as shown in Figure 1. For this semi-homogeneous solution, compatibility issues between the Vive Tracker and HMDs are avoided. However, this approach limits the choice of HMDs to Vive devices only, which can be limiting in situations where other HMDs offer important features such as higher-quality hand tracking or lower cost. The second option involved using non-HTC HMDs that may be better suited for a specific use case. However, since the Vive Tracker cannot connect directly to non-HTC standalone HMDs such as the Oculus Quest, a local PC was required to serve as an intermediary for data transfer as shown in Figure 2. This heterogeneous solution enables the tracker to be used with non-Vive HMDs, expanding the application's compatibility. Both solutions were implemented for each test case, and the following sections discuss the challenges encountered and the strategies used to overcome them. These system architectures present models for incorporating interaction devices from different vendors into an XR simulation.

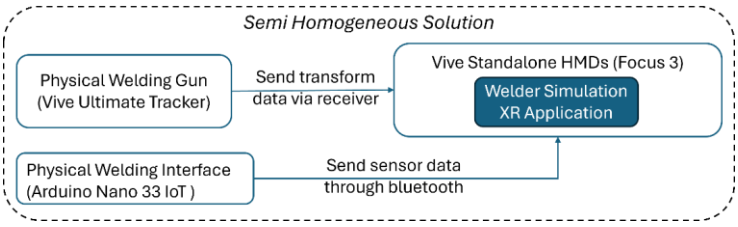

Figure 1. Semi-homogeneous System

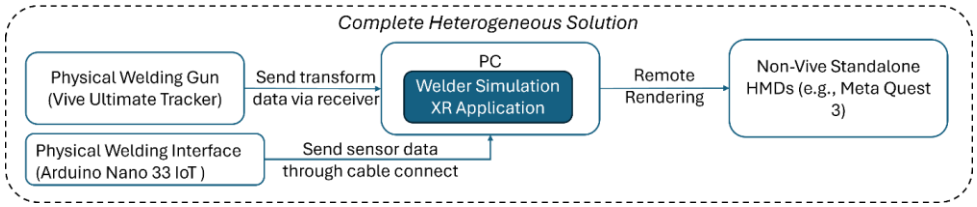

Figure 2. Completely Heterogeneous System

Case Study 1 – Welder Simulation

The first case study is an MR application designed to simulate a commercial welder. The primary design requirements include enabling users to interact with a virtual welding system, modeled after a Lincoln welder, through a physical interface and maintaining portability to demo the system in various locations. The user interactions required for this demo are interpreted from the welding machine's official user manual shown in Figure 3 (Lincoln Electric, 2005). In this procedure, steps 1, 9, 11, and 13 are virtual interactions only, there are no physical representations of the clamp or helmet. For all other steps, physical devices are held by a user and synchronized with models in the VE. To meet the design requirement of enabling physical interaction within an XR application, a simplified physical welder frame was developed as shown in Figure 4. The frame was constructed at a 1:1 scale to match the virtual welder model, using 80/20 aluminum frame and 3D-printed joints. Key interactive components such as switches, knobs, chokes, and gears were also replicated within the frame using 3D printed parts to provide passive haptic feedback. Figure 5 shows a portion of this model (left) and what a user in the VE views (right). The virtual model is mapped onto the physical model so that physical devices (i.e., knobs, switches) are registered correctly.

Welder Turn On and Off Scenario

1) Connect ground clamp to table
2) Flip the idle switch to "Auto"
3) Flip the run switch to "RUN".
4) Turn gas knob counterclockwise until flowrate reaches "25"
5) Pull chock to engage
6) Turn engine start knob clockwise
7) Push chock to disengage
8) Set output knob to "8".
9) Put on the helmet
10) Pick up the welding gun and start welding
11) Put down the helmet
12) Flip run switch to "Stop"
13) Disconnect the clamp

Figure 3. Welder Simulation Procedure

Figure 4. Welding Trainer Simulator: Physical MR setup (left) and Virtual Model registered to it (right)

To register the physical frame with the virtual model, the HMD controllers were used as positional trackers. Custom 3D-printed controller stands were mounted on top of the welder frame to securely hold the controllers for calibration, as shown in Figure 6. The forward vector of the welder frame is perpendicular to both the controller stand and the world's up vector. As a result, the forward vector of the welder frame can be calculated as the normalized cross product between the direction vector of the two controllers $(\boldsymbol{P_r} - \boldsymbol{P_l})$ and the world's up vector (U). The position of the welder frame is calculated by averaging the positions of both controllers $(\boldsymbol{P_r}, \boldsymbol{P_l})$ adding the offsets $(\boldsymbol{P_{Offset\_r}}, \boldsymbol{P_{Offset\_l}})$ from

each controller to the center of the welder frame. During the calibration process, the last three calculated welder forward vectors and positions were calculated, and the average of these values was used to determine the welder's orientation, equation (1), and position, equation (2).

Figure 5. Physical (left) and Virtual (right) Welder Interface

Figure 6. Controller Mounter

$$\boldsymbol{Q_{welder}} = \frac{\mathbf{1}}{\mathbf{3}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{I=3}}\left(\boldsymbol{P_{r,i}} - \boldsymbol{P_{l,i}}\right) \times \boldsymbol{U} \quad (1)$$

$$\boldsymbol{P_{welder}} = \frac{\mathbf{1}}{\mathbf{3}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{I=3}}\frac{\mathbf{1}}{\mathbf{2}}\left(\boldsymbol{P_{r,i}} + \boldsymbol{P_{Offset\_l}}\right) + \left(\boldsymbol{P_{l,i}} + \boldsymbol{P_{Offset\_r}}\right) \quad (2)$$

$$Q_{welder} = \boldsymbol{Forward\ Diretion\ of\ the\ Virtual\ Welder}$$
$$P_{welder} = \boldsymbol{Position\ Vector\ of\ the\ Virtual\ Welder}$$
$$P_r = \boldsymbol{Position\ Vector\ of\ Right\ Controller}$$
$$P_l = \boldsymbol{Position\ Vector\ of\ Left\ Controller}$$
$$U = \boldsymbol{World\ Upward\ Vector}$$
$$P_{Offset\_r} = \boldsymbol{Position\ \ Offset\ from\ Reft\ Controller}$$
$$P_{Offset\_l} = \boldsymbol{Position\ Offset\ from\ Left\ Controller}$$

The design of the simulation requires interaction with the physical interface inside the XR application. To support this function, sensors were embedded behind the interactive components of the physical frame to capture user input. These sensor signals were transmitted to an Arduino board (Arduino, 2025) which then forwarded the data to the XR application. To integrate the Arduino Nano 33 IoT into the XR application, the board was configured to transmit sensor data either via a wired USB connection to a computer or wirelessly via Bluetooth to an HMD. Two Arduino scripts were developed to collect input from sensors and send the data through either communication channels as needed. However, there are two major challenges in this process. First, the sensor's hardware introduced noise into the system causing unintended behavior such as the virtual gears moving erratically. Second, the high transmission rate from the Arduino overwhelmed the application resulting in frame rate reductions. To address the first issue, a denoising function was implemented directly on the Arduino to smooth the sensor output, particularly for sensitive components like gears. For the second issue, a frequency control feature was developed. The Arduino script was adjusted to send data only at a necessary minimum frequency, while the receiving system was set to ignore any missing packets when queried. This strategy reduced the communication load and improved application visual performance from less than 10 Frames Per Second (FPS) to the set cap of 90 FPS.

Another key design requirement of the simulation is enabling users to use a physical welding gun in the XR application. To achieve this, the gun must be tracked with six degrees of freedom (6DoF) and registered in the VE accurately. To make the simulation portable tracking shouldn't be done with outside-in methods requiring lighthouse base stations. The Vive Ultimate Tracker was selected, as it features an embedded inside-out depth camera that enables 6DoF tracking independently of external base stations compared with the first generation Vive Tracker, as shown in Figure 7.

Figure 7. Vive Ultimate Tracker on Welding Gun

For the HMD selection in this application, two options were considered: a Vive headset such as the Vive Focus 3 or a non-Vive headset like the Oculus Quest 3. Vive HMDs offer better native compatibility with the Vive Ultimate Tracker, non-Vive HMDs such as the Oculus Quest 3 offer other advantages such as low cost and better hand tracking. As a result, the application was developed in two versions: one for the Vive Focus 3 and another for the Oculus Quest 3. A technical evaluation was then conducted to compare their performance. However, integrating the Vive Ultimate Tracker with the Oculus Quest 3 posed challenges due to a lack of native compatibility. The first challenge arose from how to transfer data between the Vive Ultimate Tracker and Oculus Quest 3. Outside of the Vive ecosystem, the only supported platform for the Vive Ultimate Tracker is SteamVR, which must run on a local PC. To enable compatibility with the Oculus Quest, a remote rendering solution, Steam Link, was used. Steam Link allowed the application to run on a local PC, with SteamVR taking tracker input, and streaming the visual output to the Oculus Quest. The second challenge was due to incompatibility between the software development toolkits (SDKs) used by HTC Vive and Oculus Quest. To address this issue, OpenXR was used as a unifying API to support cross-platform communication. The OpenXR Mobile SDK supports native development for Oculus Quest HMDs (Meta Quest, 2025b). For the Vive Ultimate Tracker, HTC's documentation outlines a method for accessing tracker data via a VIVE Tracker Profile within the OpenXR framework. This method appears to be a viable solution when using the Vive OpenXR Plugin. However, at the time of development, this profile was not yet available without the Vive OpenXR Plugin. To overcome this problem, the existing Vive Tracker Profile for the previous-generation Vive tracker was modified to support the current Vive Ultimate Tracker. The third challenge came from incompatibility between different software required to run on PC. To connect the tracker to SteamVR on a PC, Vive Hub software was required to run on the same PC to route tracking data from the Vive Ultimate Tracker to SteamVR. Vive Hub and SteamVR needed to run simultaneously, however, launching both applications together caused Unity to crash. This issue was eventually traced to a conflict in SteamVR's input settings which was receiving hand tracking data from Vive Hub. By configuring SteamVR to not accept hand tracking input from Vive Hub, Unity stopped crashing and could run the application successfully. Compared to a homogeneous setup dominated by HTC HMDs, this heterogeneous approach enables a broader selection of compatible HMDs and opens the potential for integrating additional tools, such as Leap Motion (Ultraleap, 2025), to enhance hand tracking capabilities.

In summary, the welder simulation design required the integration of several features: 1) hand tracking, 2) customized sensors to record user input, 3) alignment between the physical and virtual welder frames, 4) object tracking for the physical welding gun, and 5) portability of the entire setup. To meet these requirements, the chosen hardware included an Arduino board with multiple sensors, the Vive Ultimate Tracker, and standalone HMDs, Vive Focus 3 and Oculus Quest 3. The different HMD options introduced different system connection structures. The first is a Vive dominated semi-homogeneous system in which the Vive Ultimate Tracker connects directly to the Vive Focus 3 and communicates with the Arduino, as shown in Figure 1. While this setup may be easier for a developer, it does present limitations such as the hand tracking capabilities built into the Vive Focus 3.

The second system is a fully heterogeneous system, where both the Vive Ultimate Tracker and Arduino are connected to a PC, which then transmits the XR content to the Oculus Quest 3, as shown in Figure 2. The heterogeneous system resolves the Vive Ultimate Tracker’s compatibility issues with other HMDs, particularly standalone devices and extends possibilities by supporting additional tools like Leap Motion and Arduino devices, giving developers greater flexibility to choose the most suitable devices based on cost, availability, and fidelity.

Case Study 2 – Veterinary Euthanasia Training XR Application

The second case study involves a simulation developed for veterinary students to practice the proper use of captive bolt guns for euthanizing cattle and dairy cows. Given that the training may take place in farm environments, the system must be portable and not rely on complex external tracking systems. The design requirements for the XR training application were formed according to inputs from subject matter experts. Given the various experience levels of veterinary students and the need to conduct training across multiple facilities, the application was designed to support multiple HMD platforms, with options for hand interaction, controller interaction, and interaction using a physical captive bolt gun. Precise alignment (i.e., position and orientation) of the captive bolt gun with the animal’s forehead is a critical component of the training, so the system requires good visual quality and accurate tracking to ensure stable and precise interactions. Based on input from subject matter experts, the following training procedure was developed, as shown in Figure 8. In this procedure, most steps are interacting with fully virtual objects, except for step 6: Aligning the gun on the cow’s forehead. For this step, depending on the setup, users can complete the task using a physical bolt gun, or a virtual one using either controllers or their hands.

Euthanasia Captive Bolt Gun Training Scenario

1. Pick up halter
2. Put halter on the cow
3. Pick up powder charges
4. Load powder charges into the gun
5. Close fire block
6. Align gun on cow's forehead
7. Pulling the Trigger
8. Check rhythmic breathing
9. Check corneal reflex

Figure 8. Euthanasia Training Procedure

Based on the design requirements, the second case study required the same physical tracking capabilities as the first one. So, two system structures were then implemented: (1) HTC Vive Focus 3 paired with the Vive Ultimate Tracker, and (2) Oculus Quest 3 paired with the Vive Ultimate Tracker via a local PC. Given the emphasis on high visual quality, the goal was to support both headsets and conduct a technical evaluation to determine which systems met the visual and tracking fidelity requirements. A screenshot of the simulation is presented in Figure 9.

A similar setup from the first case was adapted to integrate the Vive Ultimate Tracker with both HMDs to enable physical object tracking on a captive bolt gun. Unlike the first case where hand tracking was used as the primary interaction method and the welding gun could only be handled by the tracker. This application supports hand tracking, controller-based interaction, and tracker-based control of the virtual gun. However, the current setup makes Unity's XR Interaction Toolkit unable to automatically and reliably switch between hand tracking, controller input, and tracker input during runtime. To resolve this problem, an initial selection screen was implemented, allowing users to choose their preferred input method between hand tracking or controller before entering the training. When using the physical bolt gun via the tracker, another issue arose: the system continued to process hand gestures, causing the virtual gun to be grabbed by the user's hand instead of following the tracker's input. To resolve this issue, the application dynamically toggles the hand-grabbing functionality of the virtual gun depending on whether the tracker's transform is currently updating as shown in Figure 10.

Figure 9. Euthanasia Simulation Environment

Figure 10. Euthanasia Gun Alignment

Another challenge arose from the inconsistent behavior between hand and controller tracking, particularly during the release of virtual objects and locomotion. Controllers use physical buttons, allowing for precise and immediate object grabbing, releasing, and teleportation. In contrast, hand tracking relies on detecting finger joint positions to recognize gestures, making it more difficult to determine the exact moment of release which led to misalignment or unintended object placement when using hand tracking. To address this issue, the application was designed so that once users hold the virtual object steadily in place for more than two seconds, the system confirms its placement. A custom gesture-based system was also developed to enable teleportation within the VE using hand tracking.

In summary, the heterogeneous system used in this case implemented a setup similar to the first case, using the Vive Ultimate Tracker connected to a local PC with remote rendering to the Oculus Quest 3. However, unlike the first case,

this application focused more on supporting multiple interaction methods within one single XR application, enabling different interaction types based on user needs. The system architecture, shown in Figure 2, allowed all of these different interaction modalities to be seamlessly integrated into the simulation training platform.

EVALUATION

In both cases, the two different software architectures were implemented. Evaluations on each architecture were conducted from two perspectives: (1) whether the system meets the user requirements, and (2) how the technical performance compares between the semi-homogeneous and heterogeneous systems.

Case Study 1

For case one, the system needed to synchronize the user's interactions with the physical welder interface into the XR application using sensors and an Arduino board, accurately track physical components, such as the welding gun, and provide hand interaction with virtual components in the scene. The evaluation performed was not a user study, but a technical evaluation to determine if errors connecting all the interaction devices into a single XR simulation were present and repeatable. For both system architectures, the same design procedure was performed, provided in Figure 3. The procedure takes approximately two to three minutes to complete if no system errors occur. The following aspects of the technical performance of the application were recorded or observed:

1. Whether the designed procedure could be successfully completed.
2. The number of times where significant hand tracking loss or hand interaction failure occurred.
3. Frame rate (FPS).
4. Whether any visual deviations were observed, such as Z-fighting or image lagging.

Item 1 is a simple observation if a user can virtually weld with the simulator. Item 2 is a bit more challenging to determine. Loss happens when hand gestures, or hand poses, are recognized, or unrecognized, in a very short period of time that is not attributed to user input. Instead, these are unstable gesture recognitions or the inability of the system to track the hands accurately. This loss can be determined in the application in one of two ways. First, the hand joints' transforms, or controller poses, are missing, or fail to update their position or orientation when interacting with virtual objects, as is normally observed in the application (i.e., for more than 0.3 seconds). Second, the grabbing gesture, or button press, is not recognized consistently in a suitable time window (i.e., less than 0.3 seconds). Item 3 is recorded from the application itself and averaged over the simulation time. Item 4 is a qualitative measure performed by users with extensive computer graphics and VE experience, such as the authors of this paper.

Table 1. Evaluation results for Case Study 1 - Welder Simulation

	Run 1		Run 2		Run 3		Run 4		Run 5	
System	Semi-Homo	Hetero	Semi-Homo	Hetero	Semi-Homo	Hetero	Semi-Homo	Hetero	Semi-Homo	Hetero
Procedure	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Loss Events	3	1	5	0	12	0	1	0	2	0
FPS	68	90	67	90	70	90	71	90	68	90
Visual Deviations	Minor Z-fighting was observed in the background on both systems, the Oculus Quest 3 demonstrated better light rendering and a more realistic appearance.									

The result of the evaluation are shown in Table 1. For both systems, the designed procedure was run 20 times, with the results of five of those runs shown in the table. All evaluation runs exhibited the same behaviors. The welding training procedure was successfully completed in all runs (i.e., no system crashes). However, significant hand tracking deviations were observed on the Vive Focus 3, particularly during interactions with fully virtual objects such as the clamp and helmet in the welder simulation. In most runs, repeated hand tracking losses and inaccurate gesture recognition caused virtual objects to either suddenly detach from the user's hand or fail to be grabbed. These failures were counted in the evaluation as number of times hand tracking loss or interaction failed. The most notable

issue occurred when users attempted to grab the virtual helmet off their head. In this instance. hand tracking was lost when the hands moved close to the headset, making it difficult to detect the grabbing gesture accurately. Inaccuracies in joint tracking also resulted in unrealistic hand representations and gesture recognition failures leading to issues such as failed virtual object grabbing or loosely held virtual objects (i.e., the virtual object slightly moved or jiggled while being held). Both systems were configured to target a frame rate of 90 FPS. For Oculus Quest 3, the frame rate was determined by the remote rendering, which consistently maintained 90 FPS on the Oculus Quest 3 based on Steam Link. For the Focus 3, the frame rate was calculated by averaging the inverse of Unity's unscaled frame time across all frames during the procedure. The results indicate that the Quest 3 consistently achieved the target frame rate, whereas the Focus 3 exhibited a slightly lower average frame rate and did not consistently meet the 90 FPS target. Both systems showed Z-fighting artifacts in the background.

Case Study 2

For the second case, the system was designed to provide high-fidelity hand tracking and visual quality to offer users a realistic virtual practice for aligning the captive bolt gun with a cow's forehead during the euthanasia process. Considering that the application may be used in farm settings, portability and ease of setup were also key design requirements. The system was required to support both hand and controller interactions with virtual content, as well as the option to use a physical captive bolt gun for the alignment task. For both system architectures, the same design procedures, shown in Figure 8, were performed using three configurations: 1) controllers with a virtual gun, 2) hand tracking with a virtual gun, and 3) hand tracking with a physical gun. The procedure takes 1-2 minutes to complete, and the same metrics were recorded in the same manner as for case study 1.

Table 2. Evaluation results for Case Study 2 – Euthanasia Trainer

		Run1		**Run2**		**Run3**		**Run4**		**Run5**	
System		Homo	Hetero	Homo	Hetero	Homo	Hetero	Homo	Hetero	Homo	Hetero
Procedure		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Loss Events	Controller	0	0	0	0	0	0	1	0	0	0
	Hand	3	0	0	0	5	0	3	0	1	0
	Tracker	6	1	5	1	10	0	4	3	3	1
FPS	Controller	42	90	46	90	48	90	47	90	42	90
	Hand	39	90	39	90	36	90	37	90	38	90
	Tracker	44	90	36	90	40	90	38	90	47	90
Visual Deviations		Major Z-fighting was observed in the background on Focus 3, the Oculus Quest 3 demonstrated better light rendering and a more realistic appearance.									

For each configuration, the procedure was performed 20 times, with representative 5 runs shown in Table 2. The remaining runs exhibited similar behaviors. Across all configurations and runs, the designed procedure was successfully completed. In this case, the Vive Focus 3 also demonstrated lower hand tracking accuracy compared to the Oculus Quest 3. In both headsets, hand tracking loss occurred in configuration of hand tracking with the physical bolt gun tracked via the Vive Tracker. Hand tracking losses in this configuration primarily occurred when handling the physical bolt gun, as the gun often covers the user's hand. For rendering performance, the same measurement procedure as in case 1 was used. The Focus 3 had a significantly lower frame rate across all three configurations, whereas the Quest 3 consistently maintained its target frame rate of 90 FPS. This frame rate suggests that the Focus 3 may lack enough computational resources to effectively render complex, high-fidelity environment in this case. In Case Study 2, the virtual environment was more graphically demanding with higher-quality baked lighting and more detailed geometries in the scene. Some scenes in case 2 included over 1.1 million triangles in view, whereas case 1 had a maximum of 260,000. Besides the frame rate, the Focus 3 also showed noticeably poorer visual performance in Case Study 2 with significant z-fighting artifacts in the background.

Discussion

In this evaluation, technical performance differences were observed between a more homogeneous system, using the Vive Focus 3 with the Vive Ultimate Tracker, and a heterogeneous system that combines the Vive Focus 3 with the

Oculus Quest 3. In both cases, the heterogeneous system demonstrated a better technical performance in terms of rendering quality and hand tracking. For rendering performance, the heterogeneous system benefits from remote rendering via a local PC, which offers significantly greater computational power than the onboard processing capabilities of the standalone Focus 3. The Focus 3 also features a higher display resolution of 2448 × 2448 pixels per eye compared to the Quest 3's 2064 × 2208 pixels per eye (HTC VIVE, 2025a; META QUEST, 2025a). This combination of a lower display resolution and access to external computing resources may allow the Quest 3 to achieve better rendering performance and also frees up resources for tasks such as hand tracking. Hand tracking in HMDs is typically achieved by depth-sensing cameras mounted on the front of the headset, and with computer vision algorithms to calculate the position and orientation of each joint and the wrist of the user's hand. The tracking accuracy may decrease when the hand is too close to the camera, moves outside the field of view, or is occluded by a physical object, which were also observed in this evaluation.

CONCLUSIONS AND FUTURE WORK

This paper presented two use cases: 1) with a semi-homogeneous equipment setup and 2) with a heterogeneous equipment setup. Details of implementation and evaluation were presented. Evaluation results showed that both setups provided effective XR environments, from a technical perspective, with the heterogeneous setup outperforming the semi-homogeneous in some experimental runs. These findings provide evidence that heterogeneous equipment can be implemented together when designing complex XR applications for specific design requirements. It is important to note that only two setups were performed, so more work is necessary to further validate these findings. Future work will include additional system architectures exploring compatibility with other equipment vendors and potentially the development of a universal integration framework to ease the implementation of heterogeneous XR systems.

REFERENCES

Adam Vogel, M., MAJ Jeremy Rogers, U., & CPT James Sanders, U. (2018). *How can the Center for Navy Security Forces Leverage Immersive Technologies to Enhance its Current Training Delivery?* www.acquisitionresearch.net

Adriana Cárdenas-Robledo, L., Hernández-Uribe, Ó., Reta, C., & Antonio Cantoral-Ceballos, J. (2022). Extended reality applications in industry 4.0. – A systematic literature review. *Telematics and Informatics*, *73*, 101863. https://doi.org/10.1016/J.TELE.2022.101863

An, J., Lee, J. H., Park, S., & Ihm, I. (2024). Integrating Heterogeneous VR Systems Into Physical Space for Collaborative Extended Reality. *IEEE Access*, *12*, 9848–9859. https://doi.org/10.1109/ACCESS.2023.3347781

Andrews, C., Southworth, M. K., Silva, J. N. A., & Silva, J. R. (2019). Extended Reality in Medical Practice. *Current Treatment Options in Cardiovascular Medicine*, *21*(4), 1–12. https://doi.org/10.1007/S11936-019-0722-7/METRICS

Arduino. (2025). *Arduino Nano 33 IoT — Arduino Online Shop*. https://store-usa.arduino.cc/products/arduino-nano-33-iot?srsltid=AfmBOoqfUOPKHh2O7nje0CGDDGXdEQWsRaZQ56_zt56gcK9BsY_Zaq5q

Bertrand, S., Penco, L., Anderson, D., Calvert, D., Roy, V., McCrory, S., Mohammed, K., Sanchez, S., Griffith, W., Morfey, S., Maslyczyk, A., Mohan, A., Castello, C., Ma, B., Suryavanshi, K., Dills, P., Pratt, J., Ragusila, V., Shrewsbury, B., & Griffin, R. (2024). *High-Speed and Impact Resilient Teleoperation of Humanoid Robots*. https://doi.org/10.1109/HUMANOIDS58906.2024.10769949

Boyce, M. W., Thomson, R. H., Cartwright, J. K., Feltner, D. T., Stainrod, C. R., Flynn, J., Ackermann, C., Emezie, J., Amburn, C. R., & Rovira, E. (2022). Enhancing Military Training Using Extended Reality: A Study of Military Tactics Comprehension. *Frontiers in Virtual Reality*, *3*, 754627. https://doi.org/10.3389/FRVIR.2022.754627/BIBTEX

Buckingham, G. (2021). Hand Tracking for Immersive Virtual Reality: Opportunities and Challenges. *Frontiers in Virtual Reality*, *2*, 728461. https://doi.org/10.3389/FRVIR.2021.728461/BIBTEX

Caserman, P., Garcia-Agundez, A., Konrad, R., Göbel, S., & Steinmetz, R. (2019). Real-time body tracking in virtual reality using a Vive tracker. *Virtual Reality*, *23*(2), 155–168. https://doi.org/10.1007/S10055-018-0374-Z/TABLES/3

Caso, S. (2024). Emerging technologies in military space operations: current applications and future research for educational and training purposes. *International Journal of Training Research*. https://doi.org/10.1080/14480220.2024.2431482

Chen, J., & Konomi, S. (2022). Utilization of XR Technology in Distance Collaborative Learning: A Systematic Review. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13312 LNCS*, 14–29. https://doi.org/10.1007/978-3-031-06047-2_2

Chuah, S. H.-W. (2018). Why and Who Will Adopt Extended Reality Technology? Literature Review, Synthesis, and Future Research Agenda. *SSRN Electronic Journal*. https://doi.org/10.2139/SSRN.3300469

Çöltekin, A., Lochhead, I., Madden, M., Christophe, S., Devaux, A., Pettit, C., Lock, O., Shukla, S., Herman, L., Stachoň, Z., Kubíček, P., Snopková, D., Bernardes, S., & Hedley, N. (2020). Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions. *ISPRS International Journal of Geo-Information 2020, Vol. 9, Page 439*, *9*(7), 439. https://doi.org/10.3390/IJGI9070439

Garcia Estrada, J., Prasolova-Førland, E., Kjeksrud, S., Themelis, C., Lindqvist, P., Kvam, K., Midthun, O., Sverre, K., Hokstad, L. M., Mohamed, S. K., Grassini, S., & Ricci, S. (2024). Military education in extended reality (XR): learning troublesome knowledge through immersive experiential application. *Visual Computer*, *40*(10), 7249–7278. https://doi.org/10.1007/S00371-024-03339-W/TABLES/16

Gawlik-Kobylińska, M., MacIejewski, P., LebiedÅ°, J., & Wysokińska-Senkus, A. (2020). Factors Affecting the Effectiveness of Military Training in Virtual Reality Environment. *ACM International Conference Proceeding Series*, 144–148. https://doi.org/10.1145/3383923.3383950;JOURNAL:JOURNAL:ACMOTHERCONFERENCES;CTYPE:STRING:BOOK

Heinonen, H. (2023). *The Benefits of Extended Reality for Technical Communication: Utilizing XR for Maintenance Documentation Creation and Delivery The Benefits of Extended Reality for Technical Communication: Utilizing XR for Maintenance Documentation Creation and Delivery*.

Hepke, N., Scherer, M., Lohscheller, J., Müller, S., & Weyers, B. (2024). Development and Validation of a 3D Pose Tracking System towards XR Home Training to Relieve Back Pain. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*. https://doi.org/10.1145/3641825.3687746/SUPPL_FILE/SUPPLEMENTARYMATERIALS.PDF

HTC VIVE. (2025a). *HTC VIVE - VR, AR, and MR Headsets, Glasses, Experiences*. https://www.vive.com/us/

HTC VIVE. (2025b). *Unity | VIVE OpenXR - Developer Resources*. https://developer.vive.com/resources/openxr/unity/

Huang, Y., Wang, H., Qiao, X., Su, X., Li, Y., Dustdar, S., & Zhang, P. (2024). SCAXR: Empowering Scalable Multi-User Interaction for Heterogeneous XR Devices. *IEEE Network*, *38*(4), 250–258. https://doi.org/10.1109/MNET.2023.3320440

Jamah, A., Alnagrat, A., Ismail, R. C., Zulkarnain, S., Idrus, S., Mansour, R., & Alfaqi, A. (2022). A Review of Extended Reality (XR) Technologies in the Future of Human Education: Current Trend and Future Opportunity. *Journal of Human Centered Technology*, *1*(2), 81–96. https://doi.org/10.11113/HUMENTECH.V1N2.27

Kern, F., & Latoschik, M. E. (2023). Reality Stack I/O: A Versatile and Modular Framework for Simplifying and Unifying XR Applications and Research. *Proceedings - 2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct, ISMAR-Adjunct 2023*, 74–76. https://doi.org/10.1109/ISMAR-ADJUNCT60411.2023.00023

Kini, V. G., Ballae Ganeshrao, S., & Siddalingaswamy, P. C. (2024). XR Review: A Comprehensive Analysis of Visual Function Testing and Gamification in Extended Reality Environments. *IEEE Access*, *12*, 79714–79730. https://doi.org/10.1109/ACCESS.2024.3408677

Kuhail, M. A., ElSayary, A., Farooq, S., & Alghamdi, A. (2022). Exploring Immersive Learning Experiences: A Survey. *Informatics*, *9*(4). https://doi.org/10.3390/informatics9040075

Liang, H. W., Tai, T. L., Li, Y. H., & Chen, Y. C. (2022). Application of a virtual reality tracker-based system to measure seated postural stability in stroke patients. *Journal of NeuroEngineering and Rehabilitation*, *19*(1), 1–11. https://doi.org/10.1186/S12984-022-01052-0/TABLES/6

Lincoln Electric. (2005). RANGER305G OPERATOR'S MANUAL. https://www.lincolnelectric.com/en/products/k1726-5

Livingston, M. A., Ai, Z., Karsch, K., & Gibson, G. O. (2011). User interface design for military AR applications. *Virtual Reality*, *15*(2–3), 175–184. https://doi.org/10.1007/S10055-010-0179-1/FIGURES/6

META QUEST. (2025a). *Meta Quest MR, VR Headsets & Accessories | Meta Store*. https://www.meta.com/quest/

META QUEST. (2025b). *OpenXR Support for Meta Quest Headsets | Meta Horizon OS Developers*. https://developers.meta.com/horizon/documentation/native/android/mobile-openxr/

Michael Moshell. (1993). Three views of virtual reality: virtual environments in the US military | IEEE Journals & Magazine | IEEE Xplore. *Computer*, *26*(2). https://ieeexplore.ieee.org/document/192003

Min, X., Sun, S., & Qi, Y. (2024). XR CUBE: Multi-Sensory Intelligent Interaction Device for Extended Reality Application. *IEEE Access*, *12*, 78058–78073. https://doi.org/10.1109/ACCESS.2024.3401176

OpenXR. (2025). *OpenXR Overview - The Khronos Group Inc*. https://www.khronos.org/openxr/

Pallavicini, F., Argenton, L., Toniazzi, N., Aceti, L., & Mantovani, F. (2016). Virtual reality applications for stress management training in the military. *Aerospace Medicine and Human Performance*, *87*(12), 1021–1030. https://doi.org/10.3357/AMHP.4596.2016,

Peer, A., Ullich, P., & Ponto, K. (2018). Vive Tracking Alignment and Correction Made Easy. *25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings*, 653–654. https://doi.org/10.1109/VR.2018.8446435

Qin, T., Cook, M., & Courtney, M. (2021). Exploring Chemistry with Wireless, PC-Less Portable Virtual Reality Laboratories. *Journal of Chemical Education*, *98*(2), 521–529. https://doi.org/10.1021/ACS.JCHEMED.0C00954/ASSET/IMAGES/LARGE/ED0C00954_0003.JPEG

Qiu, Y., He, X., Li, Z., Peng, Z., Huang, Y., & Yu, X. (2024). Wearable Sensors for Motion and Electrophysiological Signal Tracking in XR. *Korean Journal of Chemical Engineering 2024*, 1–26. https://doi.org/10.1007/S11814-024-00227-W

Reimer, D., Podkosova, I., Scherzer, D., & Kaufmann, H. (2023). Evaluation and improvement of HMD-based and RGB-based hand tracking solutions in VR. *Frontiers in Virtual Reality*, *4*, 1169313. https://doi.org/10.3389/FRVIR.2023.1169313/BIBTEX

Rozumnyi, D., Bertsch, N., Sbai, O., Arcadu, F., Chen, Y., Sanakoyeu, A., Kumar, M., Herold, C., & Kips, R. (2025). XR-MBT: Multi-Modal Full Body Tracking for XR Through Self-Supervision with Learned Depth Point Cloud Registration. *Proceedings - 2025 IEEE Winter Conference on Applications of Computer Vision, WACV 2025*, 5041–5050. https://doi.org/10.1109/WACV61041.2025.00493

Sampige, R., Ong, J., Waisberg, E., Zaman, N., Sarker, P., Tavakkoli, A., & Lee, A. G. (2024). XR-SANS: a multi-modal framework for analyzing visual changes with extended reality (XR) in Spaceflight Associated Neuro-Ocular Syndrome (SANS). *Eye (Basingstoke)*, *38*(14). https://doi.org/10.1038/S41433-024-03147-5,

Soni, L., & Kaur, A. (2023). AR and VR in Military Combat: Enhancing Soldier Performance and Safety. *International Conference on Sustainable Computing and Smart Systems, ICSCSS 2023 - Proceedings*, 1205–1209. https://doi.org/10.1109/ICSCSS57650.2023.10169764

Steven, L., Hauw, J. K., Keane, M. B., & Gunawan, A. A. S. (2023). Empowering Military in Tactical and Warfare Area with Virtual Reality Technology: A Systematic Literature Review. *Procedia Computer Science*, *227*, 892–901. https://doi.org/10.1016/J.PROCS.2023.10.596

Theodoridou, E., Cinque, L., Mignosi, F., Placidi, G., Polsinelli, M., Tavares, J. M. R. S., & Spezialetti, M. (2023). Hand Tracking and Gesture Recognition by Multiple Contactless Sensors: A Survey. *IEEE Transactions on Human-Machine Systems*, *53*(1), 35–43. https://doi.org/10.1109/THMS.2022.3188840

Ultraleap. (2025). *Digital worlds that feel human | Ultraleap*. https://www.ultraleap.com/products/

Wang, Y., Tian, P., Zhou, Y., Zhu, M. M., Chen, Q., & Zhang, C. (2021). Using VIVE tracker to detect the trajectory of mobile robots. *Https://Doi.Org/10.1117/12.2606386*, *11884*, 531–540. https://doi.org/10.1117/12.2606386

Weichert, F., Bachmann, D., Rudak, B., & Fisseler, D. (2013). Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors 2013, Vol. 13, Pages 6380-6393*, *13*(5), 6380–6393. https://doi.org/10.3390/S130506380

Yoo, G. S., & Ji, Y. G. (2024). A Study on View Sharing AR Interface for Improving Situation Awareness during Military Operations. *International Journal of Human-Computer Interaction*, *41*(4), 2211–2226. https://doi.org/10.1080/10447318.2024.2316949;WEBSITE:WEBSITE:TFOPB;PAGEGROUP:STRING:PUBLICATION

Zintl, M., Kimura, S., & Holzapfel, F. (2024). A Mixed Reality Research Flight Simulator for Advanced Air Mobility Vehicles. *AIAA AVIATION FORUM AND ASCEND 2024*. https://doi.org/10.2514/6.2024-4651

Zoubarev, A., Bassbouss, L., Tran, D. L., Steglich, S., & Arbanowski, S. (2024). A Novel Approach for Remote Rendering and Streaming in XR. *2024 2nd International Conference on Intelligent Metaverse Technologies and Applications, IMETA 2024*, 198–205. https://doi.org/10.1109/IMETA62882.2024.10808154