Augmented Reality Mural: The Future of Collaborative Mixed Reality Design

by

Bomie Jun, Maëlle Mével, Amithi Pathak, and Hans Wassermann

Rutgers Wireless Information Network Laboratory

August 8, 2025

Abstract

The AR Mural is an augmented reality platform that enables collaborative 2D and 3D artistic creation in real-world environments. Built using Unity and deployed to the Microsoft HoloLens 2, the system integrates gesture-based input, real-time avatar tracking, and multi-user functionality to support remote co-creation. Tools such as brushes, shape creators, and object manipulators are accessed through natural hand gestures and organized via a pop-up menu. Player motion is tracked using a RealSense depth camera and YOLO-based skeletal detection, with pose data transmitted through an MQTT protocol to synchronize avatar movement across devices. Key features were successfully implemented and demonstrated, although challenges such as body-hand synchronization and hardware dependency remain. The AR Mural bridges the gap between digital design and spatial creativity, offering a novel interface for immersive, collaborative art in augmented reality.

Keywords—Augmented Reality, Unity, HoloLens, Avatar, Art

Introduction

Art is a universal expression of creative imagination that is inseparable from the human experience. However, few platforms enable collaborative 3D design with the artwork placed in the context of reality. The AR Mural fills this gap by providing a platform in which artists can sculpt both 2D and 3D artwork in augmented reality (AR), an experience that integrates virtual elements in real life through the use of special lenses. Users have access to standard design tools, and can work simultaneously with others regardless of location. This product maintains the benefits of virtual creation with online storage and fewer materials, while also letting designers see their works as tangible pieces rather than nebulous concepts. Digital media is often separate from reality, but by utilizing AR, art can be brought to life.

Background

Augmented reality (AR) combines digital content with the physical world, allowing users to interact with virtual elements in real space. This technology has seen increasing use in fields such as education, architecture, and design due to its ability to enhance creativity and collaboration in context-aware environments. Devices like the Microsoft HoloLens have made AR development more accessible by integrating spatial mapping, hand tracking, and mixed reality capabilities into a single headset [3].

To support development for HoloLens, Microsoft offers the Mixed Reality Toolkit (MRTK), an open-source framework built on Unity that simplifies the process of enabling spatial interactions and gesture recognition [2]. Unity is one of the most widely used platforms for AR and VR applications, offering a robust editor, flexible scripting tools, and native support for HoloLens deployment [5]. For remote multi-user functionality, MQTT (Message Queuing Telemetry Transport) is a common protocol that enables efficient communication between devices in real time, especially in low-bandwidth environments [4].

Additionally, skeletal motion tracking was achieved using YOLO (You Only Look Once), a real-time object detection system adapted here for identifying body and hand joint positions. Motion data was gathered via Intel RealSense depth cameras and serialized through a Python interface before being transmitted to the Unity project through MQTT.

Prior efforts in AR collaboration include systems like Spatial, which allows users to host shared virtual workspaces using avatars and voice communication. However, it is primarily focused on productivity rather than artistic creation. Another project, Tilt Brush, developed by Google, enables immersive 3D painting in VR but lacks support for real-time collaboration and anchoring in physical space. Unlike these tools, the AR Mural emphasizes both synchronous co-creation and spatial context, enabling multiple users to create and edit artwork directly within the environment it's meant to be viewed in.

System Architecture

The AR Mural was developed using Unity, a versatile game development platform that offers tools to build and launch interactive virtual applications [5]. Unity provided the foundation for designing the visual environment and programming user interactions. The scene editor allowed for quick design alterations, and by using Visual Studio Code – an Integrated Development Environment (IDE) – as the primary code editor, scripts were written and managed within the Unity environment. From Unity, the project was deployed to the HoloLens 2, the second iteration of Microsoft's mixed reality headset that overlays virtual elements onto the real world. These glasses in conjunction with Microsoft's Mixed Reality Toolkit (MRTK) enabled the system to recognize and track hand movements, meaning that gestures could be relied on for player input instead of handheld controllers. Utilizing this ability, player actions were determined by certain movements. For instance, buttons are triggered by pointing on the area in which they are displayed, and tools are active while the player pinches their thumb and index finger together.

By using a pop-up menu, users can switch between different tools. The brush tool, which operates by leaving a trail after the player's fingers while pinching, mimicking a paintbrush or pen. The line tool works in a similar manner, but instead creates a straight line with endpoints where the user started and stopped pinching. The shape tool creates different 2D shapes, namely rectangles, triangles, and circles. The object tool is similar but instead creates 3D shapes, such as prisms and spheres. The erase tool removes objects from the scene, and the select tool allows users to select and move objects in the mural. Additionally, there are undo and redo buttons that track the user's actions and can remove or restore recently made objects.

Users can contribute to multiple projects through mural selection. The AR Mural hosts many murals, each with separate drawings, similar to canvases. Before starting the program, users are prompted to select which mural they would like to edit, and users with administrative permissions are allowed to create and delete murals. The data of users' contributions are saved in a Level Database under the mural that was being edited.

A major component of the product was avatars. Because the mural is intended to provide the ability to collaborate, users have avatars that mirror their movement and motions for fellow users. This allows players to have the pertinent information about others' occupation while maintaining a degree of anonymity and without the necessity for surrounding cameras needed to import a replica of the player into AR. In order to distinguish each user, players are prompted to enter a username and choose one of eight colors for their avatar, with defaults for both fields in the event that the user decides not to enter their selections. As for the physical presentation of the avatars, the two major matters were position and movement.

Player movement was determined by a RealSense camera because of its ability to recognize depth. By placing the camera in front of the user, movements and expressions were visible. This data was serialized by YOLO, an artificial intelligence model trained to process joint positions. The software identifies twelve joints on the skeleton and eleven joints on the hand and compiles each joint's position as well as the confidence in that position. After creating a Python program with access to this data, joint positions were collected ten times per second, disregarding any coordinates with a corresponding confidence below 0.5. Sending this data from the computer running the Python file to the HoloLens required a Message Queuing Telemetry Transport (MQTT) server. MQTT is a publish-subscribe network protocol that enables connection between remote devices [1]. A broker acts as the central server that manages messages; this project used Mosquitto as the MQTT broker. Messages are published to the broker under a topic, and clients are able to subscribe to different topics, filtering what data is received. The computer with the user's serialized movement data is subscribed through paho-mqtt to the 'mural_{muralID}/avatar/heartbeat' topic which the HoloLens publishes to, and the computer publishes to the 'mural_{muralID}/avatar/pose/{clientID}' topic, which the HoloLens is subscribed to. Once the player's data is received, joint positions are connected to the avatar's rig in Unity, emulating the player's movement virtually.

Player positioning posed a significant hurdle. Users are not necessarily in the same area, meaning that their geographic coordinates could not be directly used. Furthermore, each user is assumed to use a different camera to track their movement, meaning that the relative positions serialized are not indicative of player positions relative to each other. To combat this issue, a QR code system is used such that, when scanned by the HoloLens, the user is positioned relative to a landmark in the mural, which the QR code parallels.

Results

The AR Mural prototype successfully demonstrated key features for collaborative augmented reality artwork. Core functionalities such as hand-tracked input, gesture-based tools, avatar-based collaboration, and multi-project management were all implemented and tested on the Microsoft HoloLens 2. Users were able to draw in both 2D and 3D using natural gestures like pinching and pointing, with tools responding consistently in real time. Switching between tools via the pop-up menu was intuitive and required minimal delay.

Multiple users were able to contribute to the same mural simultaneously, with joint positions transmitted from remote machines to the HoloLens using MQTT. The system maintained smooth avatar motion tracking at 10 updates per second, provided joint confidence was sufficiently high. Despite users being located in different physical spaces, the QR-code-based positioning system allowed avatars to appear anchored relative to a common mural landmark, reducing confusion about user orientation.

Preliminary feedback from demonstrations suggested the interface was engaging and the toolset encouraged creativity. However, some latency was observed in avatar synchronization under unstable network conditions, and the RealSense–YOLO pipeline occasionally dropped joints with low confidence, slightly affecting fluidity of movement.

Overall, the project met its goal of enabling collaborative artistic creation in AR, combining immersive interaction with real-time remote collaboration.

Discussion

A significant challenge encountered during development was the humanization of the avatar system. While the avatar was intended to mirror users' body movements in real time, accurately replicating human motion proved difficult—even with access to camera data. The Intel RealSense camera, paired with YOLO-based skeletal tracking, provided joint positions, but it could not capture all parts of the body at all times. Occlusion, limited camera angle, and tracking confidence thresholds meant that certain joints were occasionally missing or misaligned. Additionally, Unity's hand rig was distinct from the main body rig, making synchronization between full-body movement and hand gestures complex. This led to visual artifacts or delays in rendering expressive hand movements, which are critical for natural interaction in AR environments.

Another notable obstacle was the lack of cohesive documentation across the various technologies used. Development for Microsoft HoloLens, MQTT communication, and even the Mixed Reality Toolkit (MRTK) often required combing through scattered forums, GitHub repositories, or outdated manuals. To support future contributors and streamline development, a dedicated project website was created. This resource outlines the structure, technologies, and usage patterns of the AR Mural, providing both technical guidance and conceptual overviews.

Looking ahead, the project could benefit from replacing the current Level Database with a more scalable and flexible alternative. An ideal solution would support multi-database management, faster data retrieval, and greater compatibility with collaborative features, especially if future iterations involve more users or more complex data structures. The avatar system, while functional, remains a priority for further improvement. Future work should focus on increasing realism and consistency in avatar rendering, refining hand-body integration, and potentially incorporating facial expression tracking or inverse kinematics to enhance presence and social cues.

A key limitation of the project is its reliance on the Universal Windows Platform (UWP) due to the HoloLens being a Microsoft device. This restricts deployment to Windows-based machines, narrowing accessibility for developers using macOS or Linux. Furthermore, the recent discontinuation of the HoloLens line poses an additional constraint. As AR hardware continues to evolve, future development of the AR Mural may need to shift toward cross-platform solutions that support emerging devices and open standards in augmented reality.

Conclusion

The AR Mural presents a new way for users to collaboratively create artwork in augmented reality, offering an immersive and intuitive environment for both 2D and 3D design. By combining hand-tracking input, real-time avatar synchronization, and networked collaboration, the system enables artists to interact with virtual elements anchored in the physical world. Despite challenges related to motion tracking precision, documentation gaps, and platform constraints, the prototype achieved its core goals and validated the viability of spatially contextualized co-creation. Moving forward, improvements in avatar realism, data storage, and cross-platform compatibility could enhance the user experience and broaden the system's applicability. As AR hardware and frameworks continue to evolve, the AR Mural serves as a meaningful step toward bridging artistic expression and immersive technology.

References

[1] EMQX Team. (2025, February 26). *Mastering MQTT: The Ultimate Beginner's Guide for 2025*. EMQX. Retrieved July 31, 2025, from https://www.emqx.com/en/blog/the-easiest-guide-to-getting-started-with-mqtt

[2] Microsoft. (2022, December 15). *MRTK2-Unity Developer Documentation - MRTK 2*. Microsoft Learn. Retrieved July 31, 2025, from https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05

[3] Microsoft. (2025). *Microsoft HoloLens*. Microsoft Learn. Retrieved July 31, 2025, from https://learn.microsoft.com/en-us/hololens/

[4] Singh, A. K., & Rajput, S. (2018). An Introduction to MQTT Protocol. *International Journal of Computer Applications*, *179*(7), 38-42.

[5] Unity Technologies. (2025). *Unity Engine: 2D & 3D Development Platform*. Unity. Retrieved July 31, 2025, from https://unity.com/products/unity-engine