EE267 Final Project: Location-Anchored Virtual Reality System for Spatial Interaction and Navigation

Guilherme Simioni Bonfim

May 2025

1 Introduction and Motivation

Virtual Reality (VR) has largely matured in areas such as gaming and training simulators, but its integration with real-world geography remains underexplored. While Augmented Reality (AR) has pushed forward location-based interaction through mobile devices, the VR domain often assumes a static or abstract environment detached from physical context.

This project proposes a system where users can interact with virtual content anchored to real-world geographic locations. The design allows users to navigate spatial scenes guided by GPS data and pathfinding algorithms, while receiving real-time orientation-based feedback through a VR interface.

Inspired by early efforts to aid users with visual impairments through AR navigation, this system generalizes the concept for broader applications—ranging from accessible navigation to collaborative architecture, persistent spatial messaging, and indoor mall or airport extensions.

We further explore how IMUs, GPS (or simulated GPS), A* pathfinding, and microcontrollers (Teensy, ESP32) can work in tandem with Unity's VR capabilities to create a deeply immersive, spatially aware system.

2 Scientific Context and Related Work

The concept of persistent virtual content anchored to real-world coordinates has roots in both academic and commercial work. Early systems such as Microsoft's Hololens toolkit and mobile AR platforms like ARCore and ARKit have popularized location-anchored interactions. In the VR domain, however, spatial anchoring is still emerging.

Congdon et al. [1] introduced a method for aligning avatars across disparate spaces, creating a perception of shared spatial presence—directly relevant to our interest in collaborative extensions. Numan et al. [4] proposed SpaceBlender, which uses generative models to merge 3D environments across users. While our system does not currently integrate AI or procedural generation, the principle of composability and user-authored extensions is central.

Indoor spatial localization, such as the deep-learning mapping by Nagao et al. [2], provides scalability insights. Our simpler trilateration-based ESP32 approach aims to emulate indoor positioning affordably.

Meanwhile, Ismail et al. [3] focused on multi-user co-presence and real-time positional consistency across devices. Our architecture is designed with this extensibility in mind: paths, anchors, and environments can eventually be loaded collaboratively from a persistent server.

In contrast to past work focusing on virtualizing a space or enhancing static user experience, our system emphasizes both navigation (how users move through anchored space) and modification (what users can place there), with safety, accessibility, and collaborative design as key drivers.

3 System Design and Implementation

The full system architecture integrates Unity, VR display hardware, inertial sensors, and either GPS or simulated localization to build an immersive, location-aware navigation experience.

A* Pathfinding and Anchored Navigation

The heart of the system is a pathfinding engine based on the A* algorithm. Users navigate through a set of geographic nodes, each with latitude and longitude. Nodes are stored in a 'GeoGraph' structure with defined neighbors. The A* algorithm computes the shortest walkable path between two nodes, considering distance (via the Haversine formula) as cost.

Once a path is generated, it is visualized with Unity's 'LineRenderer', and the user is directed via floating text instructions rendered in a world-space Canvas.

Instruction Engine: Turn Detection

As the user progresses from one node to another, a 'TurnInstructionManager' dynamically evaluates the angular difference between consecutive segments in the path. Instructions are triggered based on vector angles:

- If angle $\approx 0°$ $\rightarrow$ "Go straight"
- If angle $> 25°$ left $\rightarrow$ "Turn left"
- If angle $> 25°$ right $\rightarrow$ "Turn right"

This logic runs in real time using Unity's 'Update()' method and spatial vector math, providing dynamic, directional feedback to the user.

VR Integration and Object Interaction

A VR user, rendered through Unity's XR pipeline and a Topfoison LCD headset, perceives the anchored objects and UI overlays. Head orientation is driven by a Teensy-controlled MPU6050 IMU sensor, sending yaw/pitch/roll data over USB to Unity.

User interaction is handled via a glove sensor. When the glove's input trigger is activated (e.g., via pressure or digital button), the system checks if the user is looking at an interactable object. If so, it disappears. If not, a new object (e.g., a cube or marker) spawns 3 meters ahead. This allows users to extend the environment organically and interactively.

Simulated and Real Localization

Outdoors, GPS modules can be integrated to replace simulation. Indoors, an ESP32 trilateration system is being developed. Three ESP32 boards serve as beacons, periodically broadcasting signal strength. A fourth ESP32 reads these RSSI values and calculates an estimated 2D position using signal strength-based approximation.

Unity receives the calculated position from the ESP32 via USB serial and updates the user's spatial location accordingly. This opens the door to full indoor-outdoor spatial coverage.

Graph Expansion and Anchored Content Layers

Each new virtual object is added as a node with fixed GPS or simulated coordinates. These nodes can have neighbors defined dynamically or manually. As the graph grows, the system can compute new paths and routes for future users.

The architecture is built to support layers: a base geospatial graph for navigation, and an upper content layer where users may anchor information, annotations, or models. These can represent anything from directions in an airport to a virtual museum in a public park.

Safety and Accessibility

The system is especially promising for safety-focused applications. For instance, a route could be generated for a visually impaired user that avoids stairs or unsafe intersections, simply by filtering which nodes are connected. The anchoring of virtual instructions in space, combined with tactile input via a glove, adds a multimodal layer of safety and spatial understanding.

4 Implementation Details

Unity Integration and Scene Setup

The Unity environment includes three key prefabs:

- **Node Markers:** Visual spheres placed at GPS-anchored positions.
- **Path Lines:** Rendered using Unity's LineRenderer to indicate computed A* paths.
- **Instruction Display:** A world-space Canvas with TextMeshPro shows navigation instructions.

The 'GraphDemoRunner' initializes nodes and creates paths at runtime. Each node's GPS coordinates are projected into Unity world space using a simple Haversine-based mapping and offsets from a configurable origin. A graph is dynamically constructed using a custom 'GeoGraph' class that tracks nodes, coordinates, and neighbors.

Pathfinding and Navigation Pipeline

Upon initialization, a start node is added at the player's current location. Additional nodes are created nearby to simulate spatial options (e.g., North, East, etc.), and edges between them define walkable paths. The A* algorithm calculates the shortest path, based on geodistance, and the result is rendered.

Each waypoint along the path is also added to the 'TurnInstructionManager', which evaluates the direction changes and shows "Turn Left", "Turn Right", or "Go Straight" text in the player's view. This information updates as the player progresses through the waypoints.

User Interaction with Anchored Objects and Glove Integration

An important extension to this system is the use of a custom-built glove interface. The glove includes:

- **MPU6050 IMU:** Tracks hand orientation using gyroscope and accelerometer data.
- **Pressure Sensors:** Film-based flex or pressure sensors are used on the fingertips to detect gestures such as clicks or presses.

The hardware is fully assembled and connected to an ESP32 microcontroller, with communication routed through serial. While the glove data pipeline is not yet fully integrated into Unity, the input structure is in place to allow interaction: pressing the glove while gazing at a virtual object makes it disappear; otherwise, a virtual cube appears 3 meters ahead.

The glove functionality is being implemented and tested. The hardware is fully operational; the software integration is still in development.

Indoor Localization Using ESP32 Beacons

To enable accurate indoor positioning where GPS is unreliable, we are developing a Bluetooth-based trilateration system using ESP32s.

- Three ESP32s serve as fixed-location Bluetooth beacons.
- A fourth ESP32 or Teensy device receives RSSI values and estimates position using simple trilateration.
- The estimated position is sent via serial to Unity and mapped as the user's in-VR position.

We emphasize that the beacon hardware system is already fully built and operational. The Bluetooth signal acquisition works; however, the signal filtering and accurate trilateration logic is still under testing and development.

Thus, the indoor microlocalization feature is actively being implemented: the ESP32 hardware is complete, and software refinement is ongoing.

5 Usefulness and Potential Applications

This system's design is modular and flexible, making it suitable for multiple domains:

1. Accessibility and Navigation

Visually impaired users can benefit from audio or haptic guidance using safe, annotated paths. Nodes in dangerous or inaccessible areas (e.g., stairs, escalators) can simply be excluded from the graph, ensuring safer routes. Turn-by-turn instructions and tactile glove feedback enhance orientation and spatial confidence.

2. Education and Remote Learning

Teachers could anchor content to specific rooms in a school (e.g., chemistry experiments in the lab) or build persistent, shared 3D classrooms. Students can access immersive content based on their location or navigate between lessons via pathfinding.

3. Architecture and Urban Planning

Planners and architects can place annotations or 3D overlays on construction sites. Multiple users can explore proposals collaboratively, tied to the real-world terrain, avoiding disorientation common in abstract 3D viewers.

4. Commercial and Tourism Applications

Retailers in malls or airports could use the system to place interactive objects, models, or deals tied to physical shop locations. Tourists could view historical overlays or guided routes through outdoor spaces using spatial anchors and real-world paths.

5. Emergency and Safety Systems

In emergencies, responders could be guided through marked safe zones, avoiding collapse-prone areas or blocked exits. Midpoints (nodes) in the graph could represent verified safe waypoints, and the system would only route through those.

6 Current Limitations

While the prototype is robust, several limitations exist:

- **Indoor localization** using ESP32 beacons is only moderately accurate and sensitive to obstruction or interference.
- **GPS jitter** may cause node misalignment outdoors without filtering or smoothing.
- **VR comfort** could be improved with better camera stabilization and IMU drift correction.
- **No networking** exists yet for multi-user shared views. All pathfinding and placement is local.
- **Persistence** is limited to in-session memory. Saving and loading graphs would be a next step.

7 Conclusion and Future Work

This project demonstrates a novel VR system that integrates real-time spatial interaction, geographic anchoring, A* pathfinding, and user navigation instructions. Our system allows the user to:

- View and follow anchored virtual paths in VR using head orientation.
- Receive dynamic instructions ("Turn Left", "Go Straight") as they navigate.

- Interact with spatial content using a glove-based input device.
- Anchor virtual objects to specific GPS positions and simulate movement through them.

Two major hardware-enhanced features are being implemented:

1. **ESP32-based indoor localization:** The beacon hardware is fully functional. Position estimation from signal strength is partially implemented and undergoing testing. The system is designed to support indoor environments where GPS is unavailable.
2. **Sensor glove:** The wearable glove with an MPU6050 and pressure sensors is fully built and recognized by the system. Integration into Unity interaction flow is actively progressing.

The system is robust, extensible, and practical for various applications including accessibility, collaborative design, and education.

Future work includes:

- Completing microlocalization with smoothing/filtering of beacon signals.
- Finalizing Unity input pipeline for glove-based spatial manipulation.
- Adding multi-user synchronization and shared spatial editing.
- Supporting persistent graph saving and cloud-based environment sharing.

We believe this platform opens exciting possibilities for geospatial VR and real-world integrated digital experiences. Video demos and source code are available on GitHub.

References

[1] Ben Congdon, Tuanfeng Wang, and Anthony Steed. "Merging environments for shared spaces in mixed reality." In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2018.

[2] Katashi Nagao, Menglong Yang, Xu Cao, and Yusuke Miyakawa. "Building-scale virtual reality: Another way to extend real world." In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*.

[3] Mohamad Nor'a and Mohd Nazri Ismail. "Integrating virtual reality and augmented reality in a collaborative user interface." *International Journal of Innovative Computing*, 9(2), 2019.

[4] Numan et al. "Spaceblender: Creating context-rich collaborative spaces through generative 3D scene blending." In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 2024.