An AR HUD for Navigation

Exploring how to apply AR to solve real life pain points and make navigation accessible

AYUSH AGRAWAL, University of Washington

Fig. 1. [See teaser demo video in folder]. Visual of the AR application while driving, displaying current location, destination, and the next direction to take.

As VR and AR technologies continue to develop and become more accessible to the general public, they can help alleviate pain points in our life, such as the unsafe and inconvenient modern methods of navigation while driving. I developed an AR application that users can enter their destination into, and get accurate, dynamic navigation directions projected in front of them, without needing to look away from the road, using Google's suite of geolocation, geocoding, and navigational APIs. While this novel application of technology isn't legal for a driver to use while operating a vehicle as of March 2025, we believe that future legislation around these technologies will enable us to use them across our lives, and possibly on the road as our solution does.

1 INTRODUCTION

Vehicular navigation based on directions has evolved from maps and paper directions, to directions either being displayed on a phone or screen near a car's center console away from the view of the road, or auditory directions being played over an sound system, often seconds before they need to be taken. Furthermore, for many with disabilities, navigation can be made especially more difficult and dangerous by having to rely on such methods to get from place to place. In fact, a 2017 study by AAA showed that configuring navigation systems was the most distracting activity for drivers, more so than texting while driving. [1] These applications of AR technologies in our lives, with future legislation, can be expanded far beyond just a driving HUD.

Given the regulation around such technologies, the field of AR technologies applied to navigation is quite novel. Automobile company Mercedes-Benz is working on prototypes for AR Glasses that will also display navigation details to a user as they drive. [2] Recently, some newer vehicles have begun to offer a fixed projection on the user's windshield, displaying values like the vehicle's speed and how full the gas tank is. However, these displays are quite limited, and are fixed to a small portion on the bottom of the windshield, and lack user customizability.

Thus, this project aims to fill the gaps, and proposes an AR HUD for navigation. We use network triangulation to get the user's current location as they move, computing the directions to their destination, and displaying it to them in real-time, using Google's suite of maps and navigation APIs to compute geolocation, geocoding, and real-time route determination, accounting for current traffic data.

In our results, we found that the system, does work in practice, when tested (as a passenger), driving around a local neighborhood. The successful proof of concept of such a system warrants further exploration in development of and legislation around such assistive technologies. Something we would focus on in the future is the robustification of such systems, such as the accuracy of network triangulation or other alternatives to providing an accurate representation of the current location.

Author's address: Ayush Agrawal, ayush123@cs.washington.edu, University of Washington.

Fig. 2. A Lexus RX 450's Heads Up Display

1.1 Contributions

- Several iterations with feedback from friends/family to determine and develop the most comfortable visual navigation interface for the AR display
- Research and implementation of real-time location and direction workflow, (using the Google geolocation, geocoding, and routes APIs), as well as notifications when having reached a destination. Also tested numerous times and optimized code flow to minimize latency to produce a demo with very quick updates.

2 RELATED WORK

Given the regulation around the use of most technologies while driving, the field of AR technologies applied to navigation, and any assistive technologies in the space, is quite limited to the solutions that have been available for a while.

The only other publicly announced technology in this space is a pair of AR glasses that Mercedes-Benz is working on [2]. These advanced glasses also integrate with cameras to provide a more immersive navigation interface. In a video put out by the company, the glasses appear to provide arrows on the road that point the user where to turn, with speed displays as well. However, this product is only being developed, and hasn't been commercialized yet or available to the public. We see an interface like this as the future of our display.

The most recent innovations in assistive navigation technologies has been some newer vehicle models projecting a small panel onto the bottom of a driver's windshield, as demonstrated in Fig. 2. These displays, however, are stationary on the lower part of the driver's windshield and are often bare-bones and lack customizability. The benefit of an AR headset is that the display is often larger, and moves along with the user, and can have a more customizable interface.

3 METHOD

The user can input the coordinates of their destination before beginning the drive, which provides a standard, accurate input format we can then convert to the destination address.

As the vehicle continues to move, the application repeatedly uses network triangulation to fetch the device's current coordinates, and also converts this to a human readable address, both of which we display to the user.

Additionally, every time we fetch the current location, we pass it in along with the desired destination, to compute the most optimal route from the current location to the desired destination, accounting for current traffic data. We then fetch the next step of this navigation sequence, and how far away it is. Because the distance is provided in meters, we convert it to miles when it's over 500 meters, approximately 30% of a mile, and convert it to feet for anything shorter. We display the next direction, along with the distance until it must be taken.

Finally, every time we fetch a new set of coordinates, we also compute the distance to the destination, using Haversine's formula [3] on the coordinates of both points, and let the user know they have arrived at the destination when within 50 meters of the destination coordinates.

We continued to fetch and update the current location and next direction every 2 seconds.

4 IMPLEMENTATION DETAILS

The headset that was used for the project was the HRBOX2 headset, and used a Javascript/WebGL renderer, running on a server.

When the user enters their destination coordinates, I used the Google Geocoding API [4], in order to reverse geocode the provided coordinates into a human readable address.

In order to calculate the current coordinates of the user, I used both the Javascript navigator geolocation API (primarily), as well as the Google Geolocation API, both of which use triangulation to determine the current location in coordinates. While these APIs worked to provide the current location, while driving, they weren't accurate enough to always track and update the exact position, sometimes off by a few hundred feet, which makes sense, given that the device continues to move and that triangulation is only approximate. In my video demo, I ended up mocking the coordinate fetching API (everything else was still calling the actual APIs based on the mocked coordinates) to interpolate coordinates between my start and end spot, to demonstrate the functionality if the coordinates were accurate to a few feet. I used the reverse geocoding API from earlier to also convert this to a human readable address, just so people can tell which road they're currently on.

In order to compute the directions, I used the Google Routes API [5]. I passed the origin coordinations, the destination coordinates, as well as the option to use traffic information, to the API, requesting the legs of the navigation instructions and the distances until each of them. It would then display the direction as described earlier, with units of distance determined based on the distance until the next maneuver.

Every time I fetched a new set of coordinates, I also compute the distance to the destination, using Haversine's formula, shown in Fig. 3, threshholding the distance at 50 meters as a condition of having arrived. At such a small scale, the Earth is practically flat over 50 meters, but it could scale to much larger threshholds without any imprecision.

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cos\varphi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2\tan^{-1}\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = Rc$$

- φ_1, φ_2 are the latitudes of the two points (in radians),
- λ_1, λ_2 are the longitudes of the two points (in radians),
- $\Delta\varphi = \varphi_2 - \varphi_1$,
- $\Delta\lambda = \lambda_2 - \lambda_1$,
- R is the Earth's radius (6371 km),
- d is the desired output (distance between the two points)

Fig. 3. Haversine's Formula, the distance on the outside of a sphere between two points, (Latex equations written by ChatGPT)

The application continued to fetch and update the current location and next direction every 2 seconds, based on an interval timer set to recall the workflow after a 2000 millisecond delay.

I used standard html/css for rendering the visualization, and adapted some of the code from Assignment 4 to create the stereo view that would fuse between both eyes.

One of the more challenging parts was trying to figure out how to use all the APIs correctly and setup the workflow with all the right steps, as I haven't really worked with making API calls much before.

5 EVALUATION OF RESULTS AND FUTURE WORK

The application overall ran successfully, especially when using mocked coordinate data interpolated between the different waypoints of the path. With the application, users are able to use the headset and application to provide navigation direction between any two points, even letting them know when they're reached their destination. It successfully demonstrates a working proof of concept for the space of AR applications applied to our every day lives, and also proposes an assistive technology that could help with accessibility in driving. The text and visuals were clear and readable, and fused together between both eyes.

The application solves the limitations of a stationary HUD projected onto a windshield by being at eye level of a user, that moves with them. It also provides a more visual display, that can be customized in the future.

However, the application's primary point of improvement is its robustness, which we would most likely solve with a hardware solution. The accuracy of geolocation APIs suffer from providing exact coordinate data, especially because the device is moving and running on a slower hotspot connection. Further iterations of this application would likely need a to use a micro GPS chip, that would provide accurate, real time location data, in order to overcome this issue of robustness.

The application also currently displays directions based on the cardinal directions (eg: turn East onto ...), rather than relative directions like right and left. Future versions of the application could use an API for road directional information to convert the cardinal directions into relative ones.

User studies could be conducted to determine the most important metrics that users would desire in such a HUD, for example, the current vehicular speed. Future versions of this application could add those values to the user's HUD display.

6 CONCLUSION

We demonstrated a successful AR Headset Application of a HUD for driving, providing dynamic visualizations of the directions one can take, that is quick, dynamic. This tackles the dangerous problem of safe, effective navigation while driving, and opens doors for the potential of AR technology in accessibility. It also poses questions for the future of legislation around AR technologies, and whether these technologies will be safe and helpful enough one way to warrant use while driving.

REFERENCES

[1] ACTS Law, "Do GPS Devices Increase the Risk of Car Accidents?" Accessed: Mar. 18, 2025. [Online]. Available: https://actslaw.com/gps-car-accidents/

[2] A. Upadhyay, "Haversine Formula – Calculate Geographic Distance on Earth," IGISMAP. Accessed: Mar. 19, 2025. [Online]. Available: https://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/

[3] Mercedes-Benz, "Mercedes-Benz AR Glasses Demonstration," Accessed: Mar. 18, 2025. [Online]. Available: https://www.youtube.com/watch?v=BqTGaYKLL1s

[4] Google Developers, "Geocoding API Overview." Accessed: Mar. 19, 2025. [Online]. Available: https://developers.google.com/maps/documentation/geocoding/overview

[5] Google Developers, "Routes API Overview." Accessed: Mar. 19, 2025. [Online]. Available: https://developers.google.com/maps/documentation/routes