Augmented Reality based Navigation

Aryan Agrawal (170101013)
Udbhav Chugh (170101081)
Mayank Baranwal (170101084)

CS461, Computer Graphics

Abstract

While digital maps have transformed how we approach terrain navigation, they still fall short of providing a seamless experience. Existing maps and Augmented Reality (AR) modules can often struggle to distinguish between narrow roads and streets, confusing users. This problem becomes even more profound when considering navigation for time-critical applications like ambulances, rescue operations, etc. Our AR-based navigation tool aims to provide live navigation by integrating guiding points into the user's smartphone camera experience to ensure smooth last-mile connectivity. The end-to-end Android application we have developed provides a computationally efficient solution that can run on virtually any smartphone.

1 Introduction

The project demonstration can be found *here*.

1.1 Problem Statement

How can we use AR to enable effective last-mile navigation with maps? More formally, we want to provide an augmented reality tool for short distance navigation like in streets and narrow roads where often current maps applications are difficult to interpret.

1.2 Technical Challenges to Address

Digital maps were supposed to revolutionize the way we visualize the world. Their introduction would eradicate the need for paper maps and asking people for directions. While these maps may have served their purpose in highly organized, developed countries around the world, they struggle to cope with the narrow, winding roads seen in developing countries. A quick trip to Guwahati city shows how easy it can be to misjudge alleyways and accidentally end up in the adjacent street.

Part of the problem with existing maps is that the small pointers can often cover multiple roads, making the correct road choice unclear. As such, users often find themselves very deep down a certain path before realizing their mistake. Furthermore, in developing countries, maps can struggle to keep up with the fast-changing road infrastructure. "Ghost roads," or roads that exist only in maps can often mislead users into a dead-end. Also, during natural calamities, the few landmarks that dot the road may completely disappear. It is in this context that we explore the possibility of overlaying Augmented Reality (AR) on maps to allow for smooth navigation.

1.3 Existing AR based Products

Existing GPS navigation apps primarily use 2D marking of paths (routes) on digital maps to provide navigation. The most widely used of these navigation systems is Google Maps. Recently, user location systems have been enhanced with new technology like Augmented Reality. AR is the effect of adding imaginary objects to the real physical world. The most famous and striking example of this is the Pokémon Go game. In the gameplay, a user can see Pokémons when they look at their surroundings through the smartphone's camera. However, the application of AR is not limited to games like Pokémon.

AR can also be used to enhance routing and navigation. Google Maps initially rolled Google Street View, a quasi-AR environment. Users could immerse themselves into a digitally stitched collection of photographs of the environment. However, such an approach suffered from the dual problem of providing outdated photos and no live navigation system. More recently, Google and Apple are working on beta-versions of AR-based navigation using smartphone camera for Android and iOS, respectively, but these are all proprietary products and not yet provided as a feature in their applications. Furthermore, these products works only where Google Street View is available, the problems of which were highlighted above.

1.4 AR based Navigation Research

Research in the field of AR navigation has primarily focused on car navigation and self-driving cars. One of the first papers to apply this to pedestrian navigation is [Nar+06]. In this paper, the authors first found the optimal route from third party maps. They then fed that information into a pre-computed topography map and projected a 3D model on top of it. The work's origin lies in the automobile industry and is too computationally expensive to apply to smartphones. Furthermore, the requirement of a topography map meant that only routes travelled beforehand could be utilised.

More recently, work undertaken in [CHJ16] looks at AR navigation in smartphones. Utilising Google Maps API, the authors plot the relevant points. Then, they use a fusion of sensor-based (GPS, gyroscope) and vision-based (computer vision, OpenGL) tools to project markers. However, in most cases, the waypoints tend to be too far apart to provide enough information on discerning small streets.

2 Proposed Solution

2.1 Library Utilised: Beyond AR

It is an open-source framework for creating AR android applications [Joa14] using GPS, and inserts information annotations using appropriate geo-coordinates and altitude. For this project, we have used the AR object rendering feature of this library. It helps us to create the 3D markers for a given geo-coordinate. These markers can then be plugged into a live video stream with the help of Google Maps [Gooa]. Beyond AR also provide the functionality of moving the AR object based on changes in sensor (Accelerometer/ Magnetic Field) values. This helps us to create an appropriate reaction to user movement.

There do exist other open-source AR rendering frameworks such as Google Core AR which have more diverse functionality but are computationally heavy. For our application, we only need the library to create and render AR markers. The remaining functionality is taken care of by the application itself. Thus, we chose Beyond AR as its computationally efficient and adequately served our purpose.

2.2 Approach Overview

We have built an Android application that supports live navigation based on Augmented Reality. This is our proposed solution:

- **AR Rendering:** User enters the coordinates of the destination. This can be done either by manually entering the coordinates or placing the location marker on the map. Based on the final coordinates of the destination, we render objects in augmented reality (with the aid of Beyond AR), which acts as waypoints (guiding points) for the user to navigate.
- **Marker Positioning:** Our app places the AR guiding points adjusting to the changes in the speed and direction of the smartphone. We also ensure that the rendered AR objects do not block the straight view of the user.
- **Integration with Maps:** We integrate existing maps application to find the route, which is then used to display the AR-based guiding points along the path for easy navigation.

Such a system allows the user to navigate with ease and take the correct streets through the guiding pointers. As highlighted earlier, a typical application like Google Maps can be ineffectual when handling small streets and providing last-mile connectivity. This AR-based navigation tool aids users to get proper navigation for shorter paths which current map applications have difficulty providing.

2.3 Special Features

We have incorporated some special features in our application in response to some situational challenges that may arise in the AR-based navigation which are summarised below:

- **Different Marker Features:** We use objects of different colours and sizes to depict checkpoints, alterations in direction (going left or right), etc. and changes the user's field of view.
- **Controlling Sparsity of Markers:** We have allowed the user to set how close they want these guiding points to be depending on the surrounding lighting and weather conditions.

3 Implementation Pipeline

3.1 Integration with Google Maps API

We first pass the destination and source coordinates to Google Maps API. The API provides us with the optimal path and some of the significant waypoints along the way. These waypoints include the starting and ending points along with all the turns to be taken during navigation. Most applications tend to use these waypoints with audio cues to help guide users. However, this falls short when the map instructs the user to turn left, and multiple options are available. Having retrieved the optimal path, we proceed to the next step in the pipeline.

3.2 Generating Necessary Waypoints

We use Haversine's algorithm to calculate the shortest distance between two points on a sphere using their latitude and longitude values. Haversine is expressed as:

$$haversine(\theta) = sin^2(\theta/2) \tag{1}$$

Haversine of the central angle is calculated by the following formula:

$$d = rhav^{-1}(h) = 2rsin^{-1}(h) \tag{2}$$

Solving it gives the following value of distance:

$$d = 2r sin^{-1}\left(\sqrt{sin^2\left(\frac{\Phi_2-\Phi_1}{2}\right) + cos(\Phi_1)cos(\Phi_2)sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

We divide this distance into an equal number of sections which is determined by the visibility mode chosen by the user. The number of AR markers we use depends on the visibility mode:

$$\begin{aligned} Low &: ARmarkers = (dist/3) - 1 \\ Medium &: ARmarkers = (dist/5) - 1 \\ High &: ARmarkers = (dist/8) - 1 \end{aligned} \tag{3}$$

Having split the distance into the necessary sections, we calculate the heading (or bearing) angle[Goob] using the following formula:

$$\theta = \mathrm{atan2}(\sin\Delta\lambda \cdot \cos\varphi_2 , \cos\varphi_1 \cdot \sin\varphi_2 - \sin\varphi_1 \cdot \cos\varphi_2 \cdot \cos\Delta\lambda)$$

φ_1,λ_1 *is the start point,* φ_2,λ_2 *the end point (*$\Delta\lambda$ *is the difference in longitude)*

Using the angle and distance per section, we are able to determine the exact coordinates of the intermediate markers which we use to render AR objects.

3.3 Rendering Waypoints in the 3D World

These refined waypoints, along with the generated intermediate markers, are passed on to the BeyondAR renderer. The library generates a virtual world, in which it renders these waypoints. As mentioned earlier, important checkpoints are demarcated with different symbols and colours. Additionally, marker density can be controlled as per the user's convenience as described in the previous section.

3.4 Challenges in Implementation

A few challenges we faced in the implementation of our idea:

- **Seamless Integration:** Our application needed to integrate Google Maps API, Harvesian distance calculation and BeyondAR seamlessly. Inefficient links could negate the computational advantage of our algorithm.
- **Computational Cost:** The choice of our marker's positioning algorithm must work on low-end smartphones and only require GPS, camera and Internet connection and be computationally inexpensive.
- **Android Studio:** While not our first experience with Android Studio, the environment presents many challenges. In particular, the new framework, API support and structure are different from standard applications.

4 Limitations and Future Work

Though our project presents an end-to-end solution, some additional features can be incorporated if we take it further:

- **Offline Navigation**: Our application currently requires an active Internet connection to function. We could potentially cover a brief drop in the Internet connection with an accelerometer. However, the current technology is under-developed and requires significant development to be accurate.

- **Marker Aesthetics**: The markers' highlights may not reflect the lighting direction of the environment. This is done to allow for a low-computing solution but can cause some difficulties to the user.
- **Occlusion**: To save further on computational power, we can use a periodic background thread to continually update the current location and not display occluded markers.
- **Haptic Feedback**: We can add vibrations to guide the user, especially at turns or if the user is going down the wrong path. This can make the AR experience immersive and could also help out blind people who can't take advantage of the visual features of the app.

5 Contributions and Conclusion

The 3 of us equally contributed to the research and development of the AR-based Navigation application.

Navigation has come a long way from paper to digital maps to now AR. AR-based navigation is the latest frontier, as it can resolve a large number of problems encountered with digital maps. Our end-to-end Android application provides a lightweight and real-time AR navigation experience to enable better last-mile connectivity. It can also be applied in the field of disaster management when streets cannot be differentiated. By integrating Google Maps, Haversian distance and BeyondAR, we make AR navigation available to the cheapest of smartphones.

References

[CHJ16] Chee Chung, Yilun He, and Hoe Jung. "Augmented Reality Navigation System on Android". In: *International Journal of Electrical and Computer Engineering (IJECE)* 6 (Feb. 2016), pages 406–412. `https://doi.org/10.11591/ijece.v6i1.9345`.

[Gooa] Google. "Android Studio Maps Documentation". In: (). `https://developers.google.com/maps/documentation/android-sdk/overview`.

[Goob] Google. "Spherical Geometry Concepts in Android Studio". In: (). `https://developers.google.com/maps/documentation/javascript/geometry`.

[Joa14] IagoLast Joan Kevin Kasamo. "BeyondAR Framework". In: (2014). `https://github.com/BeyondAR/beyondar`.

[Nar+06] Wolfgang Narzt, Gustav Pomberger, Alois Ferscha, Dieter Kolb, Reiner Müller, Jan Wieghardt, Horst Hörtner, and Christopher Lindinger. "Augmented reality navigation systems". In: *Universal Access in the Information Society* 4 (Mar. 2006), pages 177–187. `https://doi.org/10.1007/s10209-005-0017-5`.