SMART CITY SOLUTIONS FOR LOCATION TRACKING

Neetesh Bhati, Sujala D. Shetty\(^1\) and Aparna Raj

\(^1\)Dr.
Department of Computer Science,
Birla Institute of Technology and Science, Pilani, Dubai Campus, U.A.E

ABSTRACT

With the advancement in technology, cities are getting smarter and becoming more livable and responsive. In the next few years, we expect a significant rise in the use of data and technology to make better decisions and improve the quality of life. Outdoor navigation using GPS is relatively easy but finding way indoors is still considered as a frenetic task. We aim to use data from multiple sources to create a detailed profile of the geographic information for every individual which can be easily used and securely accessed whenever needed. In this paper, we propose a method for indoor navigation which guides the user to a specific spot using smart phones. Here A* algorithm is used to find the shortest path between two points on the map and ArcGIS software for creating the map data. The generated map along with the required services are hosted on Osiris, an open-source indoor map rest API server. The server is connected to a front end using an Android application developed with Android Studio. This can be applied across large companies which are situated in sprawling campuses.

KEYWORDS

GIS, Location Tracking, Indoor Navigation, Android Application

1. INTRODUCTION

Navigation is something that people have been doing since the early days and in modern times, it has continued to develop for the outdoors with the aid of technology such as GPS (Global Positioning System). Even though it provides incredibly precise positioning, it is not accurate for indoors as it relies on satellite line of sight. Also, the main reason that lacks in obtaining the accurate indoor positions is the difficulty in creating indoor maps with minimal infrastructure.

With indoor location and navigation being a hot subject, we are trying to investigate how it can be done to a functional degree with Wi-Fi routers and ideally expanded from there by the infrastructure. This subject is being addressed by many companies including Microsoft, who have grown it into an actively developed area. Hence our goal is to develop a small version of the indoor navigation framework for Android. For the easy deployment of the system, we make use of ArcGIS platform which provides the location-based analytics.

2. RELATED WORK

Providing an indoor navigation system has its challenges. The first and foremost essential factor is the accuracy in determining the position of the device. The next challenge is in mapping the interior of the location and storing it in the database for easy access. Finally, we must find a path between the two points to display it to the user. The path must be shortest, fastest, and easiest to take. Determining this path has its computational complexity (Poply & Shetty, 2019). Though there are multitude of methods such as lateration, fingerprinting, dead reckoning etc. none of them provides actual shortest route to the user (Jackermeier & Ludwig, 2018) (He & Chan, 2016) (Palamakula, 2015). Some of the existing mechanisms are described as follows.
In (Jamil et al., 2020), authors used RSSI (Received Signal Strength Indicator) trilateration method for determining the indoor position. It is an estimated measure of power level that an RF client device is receiving from an access point or router. This ensures that the system can get the signal from the nearest point of entry called Access point (AP), which is connected to a wired router, switch, or hub via an Ethernet cable, and projects a Wi-Fi signal to a designated area. The position of the recipient is then determined by these distances and AP co-ordinates. But, because of the multipath and non-line of view (NLOS) propagations, the precision and reliance on one-point positioning were low because of the non-stationary character of the RSSI.

Enhanced alteration methods based on RSS are implemented in (Wang, 2011). The paper includes a framework focused on regression and a method based on correlation. The regression-based approach uses linear regression to depart from the distance between the wireless system and the access point as well as the associated signal intensity reading from the propagating model. The correlation-based method has been used to match the long-distance model with correlated RSS measurements from a local location. The regression approach led to reducing the error for the Wi-Fi network from 21 feet to 15 feet while the correlation approach decreased the error further from 22 feet to 12 feet.

In (Satan, 2018), authors triangulate the position of a device with the aid of the signal strength of several receivers. In (Brena et al., 2017), authors used an RF method. They introduced a few methods of optimizing precision, but they found that the use of the NNSS (Nearest Neighbor in Signal Space) performed well when applying to search space collected offline, which provides an outline resolution of 2-3 meters in length.

3. DESIGN METHODOLOGY

Indoor navigation with Wi-Fi signals existed as early as the early 2000s, is no invention. In the field of device development, i.e., the real deployment, there have been several attempts to achieve it, including the Microsoft Android Route Guide, which only received 10,000+ downloads (which is in the 10,001 - 50,000 range). Also, indoor plans of certain buildings are supported by Google Maps but do not provide indoor navigation. Another example is a site mapping application that can only use cell data, and not Wi-Fi, for indoor navigation. This program has only received over 1,000 (1,001 – 5,000 range) downloads. So, we took on this challenge to create an app for indoor navigation.

There are several components to keep in mind while designing a full stack application. It includes deciding the backend server, the frontend application client, the database server schema, the API structure which will be used to facilitate communication between the several components of the application, etc. In Figure 1, we have laid out the basic architecture of the application. It has an API for communication with Android application, web server, and data storage.

![Figure 1. The application architecture](image)
We have used several APIs that help us to overlay the floorplan image on a map with geo-tags (GeoTIFF) using data that we had previously prepared. The next move is to incorporate user location and route scanning. However, we soon discovered that in the MapBox API we could not explicitly track indoor route seeking (or that would take way too long). The MapWize API is developed on MapBox itself. We were presented with a system for overlaying a pre-processed floorplan on a real-world map, inserting our space polygons, entering additional room details (like the applicable search terms for users, for instance: 'Place', 'Class' and '267'). The wayfinding had to be carried out by grouping various nodes between rooms over the whole floor so that potential roads could be diagramed. Besides, we were able to download the WiFi router’s locations in a JSON file, which was useful at the next stage, which was to get the individual user’s location, by entering every router and noticing details and place.

4. INDOOR NAVIGATION

4.1 The Navigation Algorithm

A* is like Dijkstra Algorithm for determining the shortest path. A* can be thought of as the first Greedy search and it can direct itself by using a heuristic. It is as fast as Greedy approach algorithms in a simple case. The key to its effectiveness is that it blends the pieces of information used in the Dijkstra algorithm with the information used by Greedy Best-First-Search (favoring vertices that are close to the goal). G(n) is the exact cost of the path from the starting point to any vertex n in the standard language used in A* and h(n) the approximate heuristic cost from the n vertex to the target. The yellow (h) is far from the goal and (g) is far away from the start (Sifat Hassan et al., 2020). A* balances the two as they pass from start to finish. The vertex n which is lower f(n) = g(n) + h will be tested each time through the main loop (n).

Figure 2. represents the shortest path calculation between two specified points (Aydin et al., 2017).

Table 1. Different algorithms for shortest path calculation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Comments</th>
</tr>
</thead>
</table>
| A*        | Based on Heuristics  
|           | G - Exact cost of path from initial node to current node  
|           | H - admissible cost to reach the goal from current node  
|           | F = G + H  
|           | If H is overestimated, the goal node is found faster.  
|           | If H is underestimated, best results but longer processing time. |
| BFS       | Graph G = (V,E)  
|           | Algorithm complexity in time depends on O(|E| + |V|) when every edge and every nodes is visited.  
|           | In reality, O(|E| + |V|) can vary between O(|V|) and O(|V|^2) depending on graph edge evaluation. |
| Dijkstra’s| THE algorithm execution time is O(|E| + |V|^2) and in worst case the performance is O(|E| + |V| log |V|) |
| HPA*      | Internal edges in a cluster is obtained by running A*  
|           | HPA* is 1% optimal path finding algorithm |
| LPA*      | Ideal for changing networks, derived from A* |

Table 1 compares various algorithms along with their merits and demerits which are suitable for path finding application. We have tested these algorithms via a simulation to find the most efficient one for this paper. Since the proposed algorithm computes amongst lesser nodes, optimized network is deemed more efficient (Apanaviciene et al., 2020). It often discovers the next best route also. A* maintains a priority queue of options that it’s considering, ordered by how good they might be (Kotaru et al., 2015). It keeps searching until it finds a route to the goal that’s so good that none of the other options could possibly make it better.
4.2 The ArcGIS Maps

There are various components on the ArcGIS platform which are communicated through the ArcGIS REST API and common file formats. To create a good application for ArcGIS, knowing the common elements of the software is important. ArcGIS Online and ArcGIS Enterprise are central to the ArcGIS network. The platform hosts GIS services and a gateway with online UIs and APIs that allows content, maps, apps, and users to be written, posted, and managed. We can interactively use the portal or provide direct access to the underlying REST facilities (Alam et al., 2018).

In addition to ArcGIS Online and ArcGIS Enterprise, ArcGIS Pro and ArcMap are part of the desktop suite that allows users to publish and manage data and charts. These desktop applications can also be expanded and written using multiple tools for comprehensive and advanced analysis. The ArcGIS Desktop website provides a rundown of the features of the ArcGIS Desktop.

There are several data formats supported by ArcGIS:
1. Shapefile – Esri’s semi-open, hybrid vector data format using SHP, SHX, and DBF files. Originally invented in the early 1990s, it is still commonly used as a widely supported interchange format.
2. Enterprise Geodatabase – Esri’s geodatabase format for use in a relational database system.
3. File Geodatabase – Esri’s file-based geodatabase format, stored as folders in a file system.
4. Personal Geodatabase – Esri’s closed, integrated vector data storage strategy using Microsoft’s Access MDB format is a legacy format generally replaced by the file geodatabase in most contemporary use.

4.3 The Hosting Web Server

Osiris provides a solution for indoor mapping and APIs for indoor access to GIS data. This is developed as an easy-to-install, ready-to-use app package for small to medium projects. The building data is spread as a
vector for the creation of apps that represent the pages as four of the key features of Osiris. Osiris has an additional Interest Points API to show external outside positions. Figure 3. & 4. shows the design of an ArcGIS Map and how it is hosted on a web server.

There is a build.sh script which creates all the libraries and the “fatJar” files needed. After executing it, we can find a collection of jar files and scripts within the bin directory. This folder can be used as a sandbox for trying Osiris. Also the configuration files EnvConf.yml and env.properties provides the customization of the installation. There are two things to do to set an Osiris environment. First, we need to import a map, second, we will launch our Rest Services application. For both processes a MongoDB instance should be running. This is represented in Figure 5.
4.4 The Front-End Application

Every web server or a backend API requires a frontend application to display data. In our case, we have created an Android application that connects to the web server using the preconfigured API and displays the map data. The Android application is created using the Android Studio as represented in Figure 6. It is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. Figure 7. represents the android app. created with REST API server and MongoDB for indoor navigation.
5. CONCLUSION

In this paper we focus on the task of indoor navigation particularly to provide precise user location and to deliver shortest path between user specified locales. We have created an android application with REST API server running with a MongoDB database to store the data required. Overall, a successful result with thorough room navigation was achieved. The only time it failed was that the closest routers were not properly changed when the spot on the map shifted marginally from the current spot. Yet we strived best that could be achieved, keeping in mind the shortcomings of Android OS itself. By looking at improved methodologies, the accuracy of localization itself may be improved by researchers to achieve decimeter-level precision. This idea could be turned into a modular system, in which organizations can employ things like floor plans and indoor Wi-Fi router locations, i.e., taking advantage of their data. Perhaps it could be applied across an institution particularly the ones that are spread across multiple campuses.

We expect this technology to be implemented shortly with better algorithms, hardware, and software to make the internet a safer place. Also, if any chances of lessening the rate of access point scans, it could lower the deployment cost. Henceforth, the overall accuracy can also be improved even though the project is applied in the same manner. Moreover, there are several different indoor navigation techniques, we can use either one of those to test and find the accuracy and then implement if we are satisfied with the results.

REFERENCES


