Location Tracking and Smooth Path Providing System

Smita Jangale[1], Rutuja Rajhans[2], Anuja Shetye[3], Ritik Wadhwani[4]

[1]Professor, Dept. of Information Technology, Vivekanand Education Society's Institute of Technology, Mumbai University, Mumbai, India
[2,3,4]Student, Dept. of Information Technology, Vivekanand Education Society's Institute of Technology, Mumbai University, Mumbai, India

---

Abstract - *The project paper entitled "Location Tracking and Smooth Path Providing System'' focuses on the objectives, methodology and the implementation of the created system. As the GPS signals are not that reliable due to some issues on low-frequency data; it is a challenging task for finding real-time location coordinates and getting an accurate travel route. So, a solution was proposed for catering the goals of the project by making use of the Machine Learning and Data Science Algorithms. Literature survey was done through some of the research papers of IEEE, Springer, ScienceDirect which provided a precise information and view regarding location measurement, turn analysis, route analysis methods, computing travel time and algorithms like Dijsktra's Algorithm for providing with the shortest route and Kalman filter for smoothening of the track. In some way, the system will have a huge impact on traffic management, and would also help to cater the needs for providing precise paths thereby benefiting the people and would reduce their efforts to travel an extra mile. The system is user friendly and provides results in good deliverable time.*

***Key Words*: Low frequency data, location measurement, turn analysis, route analysis methods, Kalman filter and smoothing algorithm, API.**

1. INTRODUCTION

In this modern era, vehicles are fully dependent on GPS service's accuracy and digital road map quality instead of physically searching for the road routes. Even with the most up-to-date mapping and navigation software, GPS tracker is still at the mercy of its satellite network. Accuracy problems can arise from atmospheric to terrestrial. When a satellite isn't able to transmit its position due to orbital error, it won't be able to establish a link with a GPS tracker.

Here comes the need for implementing a convenient and handy system for real-time location finding and getting a smooth and accurate path to destination. Also in the current scenario, for traveling from a source point to a destination point, one has a lot of routes to choose from, so which one to choose from, so as the overall route is shortest and the travel time is reduced. To address this, an idea was thought of implementing a project titled "Location Tracking and Smooth Path Providing system".

1.1 RELATED WORK

GPS is used for tracking vehicles and keeps regular monitoring. Atmospheric conditions including temperature, pressure, and humidity can cause calculation and accuracy errors in the satellite network. To tackle this, a research paper[4], proposed a path planning approach which refers to finding a collision free and feasible path from the given initial to final configuration and reshaping path, along with a detailed approach of graph search and optimization based algorithms. Also, a paper[5] published gives deep insight into Dijsktra's algorithm. It provides the approach for resolving the surface optimal path planning using extended Dijsktra's algorithm. Researchers had deployed various map matching algorithms which combined the concepts from Kalman Filter and advanced map matching algorithms for providing path at every curvature using the best segment selection approach based on the information about weight of closeness, direction difference which resulted in processing less amount of data[6]. A dynamic approach to predict travel time in real time using data driven techniques and comprehensive data sources provided insightful information regarding mathematical formulas for computing travel time required for finding the optimal path[7].

1.2 PROPOSED SOLUTION

The problem statement is "Location Tracking and Smooth Path Providing System". The project aimed at delivering an optimized route from source to destination by tracking the real time coordinates, displaying nearby amenities and facilities through the route, weather conditions and thereby providing the approximate travel time for the same. Also, sharing of current location with others in case needed, is also embedded in the project.

Initially, the source and destination address for which user wants to travel the route has been provided by the user. By using various libraries of python, tracking of all the possible routes and the optimum amongst them was done using data science techniques. Dijkstra's Algorithm was used for getting the locality of points of the shortest route and Kalman Filtration Technique on those points was applied to get a clean and smooth path on the map. Scattermapbox was used as the means of representing the route over a map. Nearby amenities, weather and

temperature conditions and current location of the user were fetched through API. Travel time was predicted so the user can decide his journey beforehand.

2. PROJECT METHODOLOGY

2.1 SOFTWARE REQUIREMENTS

- Google Map API - for retrieving current location
- OpenWeatherMap API - for retrieving weather and temperature data
- OpenCage API - for geocoding
- Google Colab - platform for making the system
- Flask - for deploying the system

2.2 HARDWARE REQUIREMENTS

- RAM on Colab notebook - 0.77 GB/12.69 GB
- Disk Space on Colab notebook - 46.32 GB/107.72 GB

2.3 PYTHON LIBRARIES USED

1.Numpy - It provided a multidimensional array object, as well as variations such as masks and matrices, which were used for various math operations.

2.Plotly - The plotly library is an interactive, open-source plotting library which was used in plotting the route over map.

3.flask-ngrok - This library was used to make the url of the web app public so that it can be used with colab.

4.OSMnx - It was used for downloading geospatial data from OpenStreetMap and model, project, visualize, and analyze real-world street networks and other geospatial geometries.

5.NetworkX -It was used for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

6.Jinja-2 - It was used for providing a web template engine for the system.

2.4 TECHNIQUES/ ALGORITHMS STUDIED

1. Dijkstra's Algorithm - Dijkstra's algorithm is the iterative algorithmic process which was used to provide the shortest path from one specific starting node to all other nodes of the graph.

2. Kalman Filter and Smoother - Kalman filter algorithm is an iterative mathematical process which was used to detect and estimate the true value, position, velocity, etc. of an object or motion of an object, when the measured values had unpredicted, error or variation.

3. Map Matching Algorithm - A link for a smooth and reliable path on the road was provided by a map matching algorithm.

4. Euclidean Algorithm - It was used to get the nearest or neighbourhood point from the desired point for accuracy without altering the process.

3. BLOCK DIAGRAM

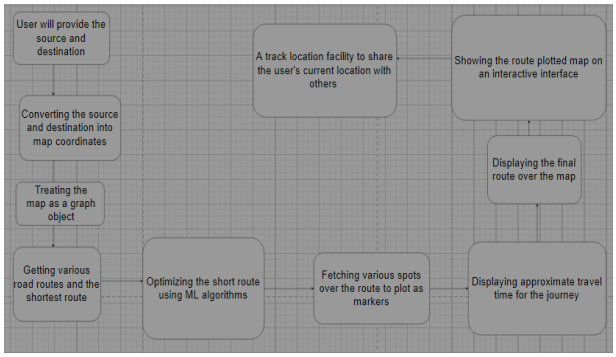

Fig -1: Block diagram of the system

4. IMPLEMENTATION

A) Getting the source and destination address from the user

The system was deployed using Flask which is a Web Framework in python for providing user friendly and convenient interface access for the user.

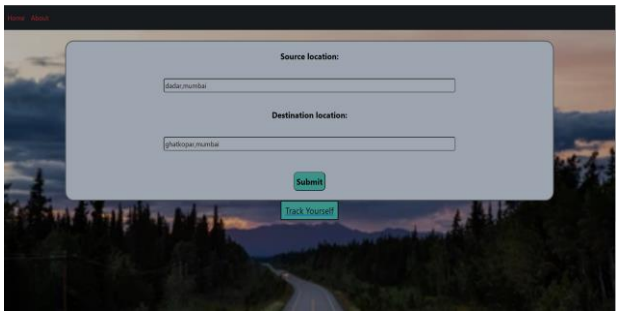

Fig -4.1: Home Page

OpenCage Geocoder API - The OpenCage Geocoding API provides accurate reverse (latitude/longitude to text) and forward (text to latitude/longitude) geocoding via a RESTful API. The user provided physical address was converted into map coordinates using OpenCage. Accurate existing address was the necessary condition for getting the next steps to be done.

Google API - Google APIs are application programming interfaces developed by Google which allow communication with Google Services and their integration

to other services. The track yourself facility used the API for getting the current location of the user. Below is the map file that gets generated which can be shared with others.

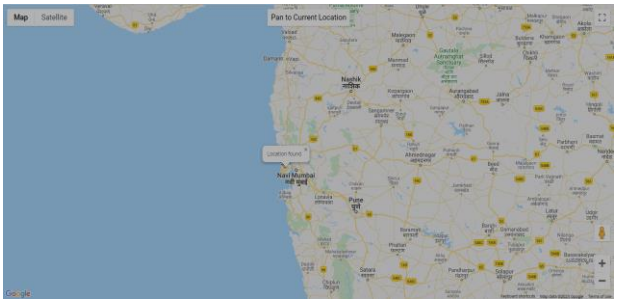

Fig -4.2: Map file of user's current location

B) Processing on the obtained coordinates from user

Many libraries can be used to plot a path using various APIs, but this leads to reduced flexibility. Also, on many occasions, a user may want some flexibility of not only having one path between two points but also multiple paths which he can select from, for example: user wants to look for all the existing routes beforehand.

a. Visualizing the graph view of road network between the source and destination -

Visualizing was done using the OSMnX package.

OSMnX - OSMnx downloads the map as a graph object which can easily be used by the NetworkX library. So, with the coordinates obtained from OpenCage, the graph structure of the road network was visualized. The graph view was not just a stationary image but it had a lot of information that was used for plotting routes. The edges or lines in the graph denoted the roads and the nodes or points in the graph denoted endpoints of the roads. Also, the edges and nodes were associated with unique id, map coordinates, geometry, and name so that they can be used for plotting over the map. The nearest node was obtained to source point and destination point respectively using the get_nearest_node function by Euclidean's Algorithm of OSMnX for plotting routes between them as the exact coordinates were not available on the graph.

Below is the illustration of the graph object.

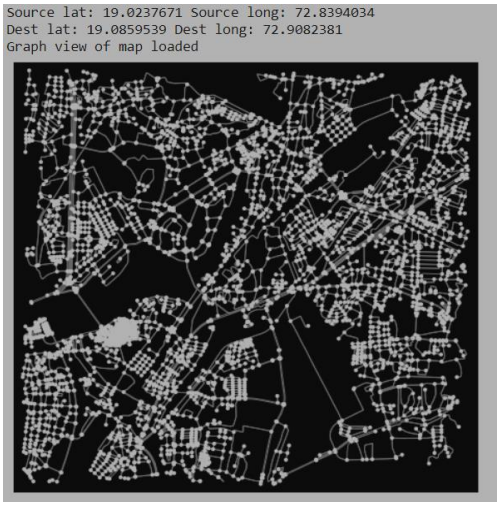

Fig -4.3: Graph structure of the road network

b: Getting all the possible routes and the shortest one -

Possible routes and the shortest one was obtained through the NetworkX package.

NetworkX - The k_shortest_paths function of OSMnX retrieved all the possible paths and the shortest_path function of NetworkX implemented Dijkstra's Algorithm internally for obtaining all the points from start node to end node from the graph such that the path was shortest. The shortest_path _length function provided the approximate travel time in minutes. So, after getting all the routes, the latitude and longitude of all the nodes to be plotted on the map was stored.

From the below figure, all the existing routes can be visualized on the graph before plotting them on the actual map. The nodes are connected together to form a path.

Fig -4.4: Red paths denoting all the existing routes

The below graph depicts the shortest path.

Fig -4.5: Shortest route by Dijkstra's Algorithm

c: Kalman Filter estimation on the shortest route

The next step after getting the shortest path was to simplify and smoothen the path.The goal was to reduce the number of points which were quite inconsistent on the path to get a precise and optimized path without any inconsistency. This was done with the help of Kalman Filter and Smoother Algorithm.

The Kalman Filter Algorithm is an iterative mathematical process. It detects or estimates the true value or position of a point when the measured value contains unpredicted, error or variation. This algorithm was estimated on the latitude and longitude of the shortest path. Two algorithms for tracking were implemented : the Kalman Filter and Kalman Smoother with the help of Pykalman module of python. Three filters were used and finally points of the shortest route to be plotted over the actual map were obtained.

Kalman Filter	Description	Output
I	It took an initial guess of state estimate and initial guess of error covariance matrix as input and transition matrix where last state of nodes was updated to reflect the current state and observation matrix where current state of nodes was transformed to the best estimate of the system.	It did a reasonable job of rejecting noise.
II	It took an extra parameter i.e. observation covariance and to avoid reestimation it was fixed to 1000 times to that of the first Kalman filter observation covariance.	A smooth estimated state of nodes was obtained.
III	Here the final filter was used on the line.	The plot considered the points and provided a path through them too. The average Time to build and train Kalman Filter III was 0.9103 seconds and the updation process of Kalman filter III was also carried in 3 stages.The average time of first stage of the updation process was 0.0008659 seconds, second stage required 0.00146 seconds and third updation stage required 0.00111 seconds.In this way Kalman Filter estimation was done.

Table -1: Kalman Filter & Smoother estimation

From the below three figures, the blue line depicts latitude coordinates and red line depicts longitude coordinates.

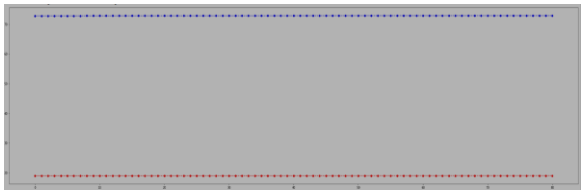

Fig -4.6a: I Kalman Filter

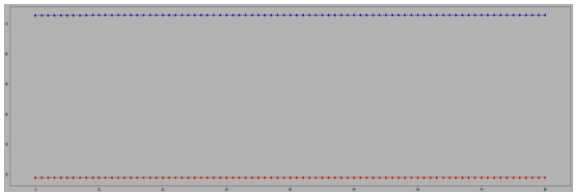

Fig -4.6b: II Kalman Filter

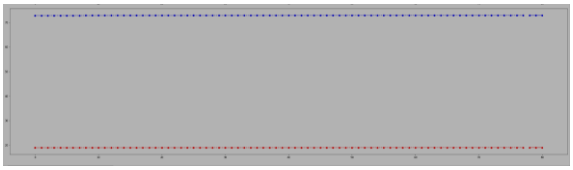

Fig -4.6c: III Kalman Filter

d: Fetching various amenities and facilities through the optimized route -

Opencage API was used for picking out the coordinates over the optimized route which represented an amenity like restaurant, shop, cinema, cafe, education related institutes like schools, colleges and clinics, hospitals for plotting them as markers.

e: Getting weather and temperature conditions -

OpenWeatherMap API was used for this purpose. It is an online service, owned by OpenWeather Ltd, that provides global weather data via API, including current weather data, forecasts, nowcasts and historical weather data for any geographical location. So, the temperature and weather data was retrieved from the API for the source and destination address.

C) Plotting the final route on Map

Scatter Map was used for plotting purposes. Scatter Map allowed to visualize geographical data as data points on a map. The Plotly library plots all the nodes and connects them with lines to represent a path. So after collecting and compiling everything together, the final route was plotted on the map along with the multiple routes, travel time, nearby amenities, weather and temperature data, source, destination and notations. The below figure is the snapshot that the user views after clicking on submit.

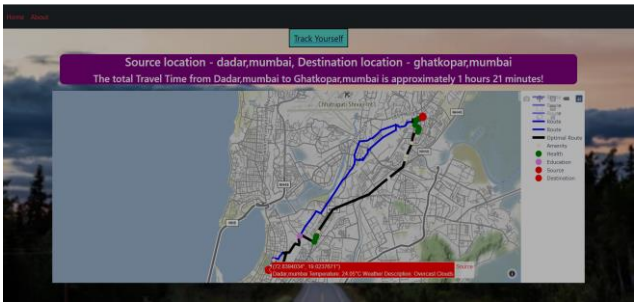

Fig -4.7: Final map seen by user

5. RESULTS AND DISCUSSIONS

In some way, this system will have a huge impact on traffic management, and would also help to cater the needs for providing precise paths thereby benefiting the people and would reduce their efforts to travel an extra mile. The system was user friendly and provided the results in less deliverable time. The time taken to build and train the Kalman Filter was 1.412 seconds which would be further improved to give a better accuracy. The main objective to have an accurate path selection process on appropriate road networks and design an approach which minimizes storage requirements while not compromising on the prediction accuracy was achieved.

6. SCOPE

To provide the user with the accurate results in less deliverable time is the scope of the project. The system is quite user friendly and interactive. The scope boundary is that the user needs a good internet connectivity and a smart device. No need to login using the system.

7. CONCLUSION

Thus the project entitled "Location Tracking and Smooth Path Providing System" was successfully implemented and developed by taking care of all the possible constraints. The average waiting time for the system to provide the results is satisfactory.

8. APPENDIX

Definitions :

1. API - API is an acronym for Application Programming Interface that software uses to access data, server software or other applications. APIs are very versatile and can be used on web-based systems, operating systems, database systems and computer hardware.

2. Flask - Flask is a web framework. It provides tools, libraries and technologies that allows one to build a web application.

3. Google Colaboratory - "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

ACKNOWLEDGEMENT

We would like to thank our mentor, Mrs. Smita Jangale for guiding us and enabling us to make sense of all the information regarding the project and also thankful towards the authors of the papers we have referred for providing us with the better understanding of the problem statement and grateful towards everyone.

REFERENCES

[1] Bonnifait, P., Laneurit J., Fouque C., D'herbomez, G., "Multi-hypothesis map matching using particle filtering", In: 16th World Congress for ITS Systems and Services. pp. 1–8 (2009)

[2] Aric A. Hagberg, Daniel A. Schult, Pieter J. Swart, "Exploring Network Structure, Dynamics and Function using NetworkX", proceedings of the 7th python in Science Conference (SciPy 2008)

[3] Goh C.Y., Dauwels J., Mitrovic N., Asif M.T., Oran A., Jaillet P., "Online map-matching based on hidden markov model for real-time traffic sensing applications", In: 2012 15th International IEEE Conference on Intelligent Transportation Systems. pp. 776–781. IEEE (2012)

[4] Chaoyi Sun, Qing Li, Li Li, "A grid map path reshaping algorithm for path planning", In:2019 IEEE major project "New generation Artificial Intelligence" under China.

[5] Min Luo, Xiaorong Hou, Jin Yang, "Surface optimal path planning using an extended Dijkstra Algorithm", In:2020 IEEE volume 8.

[6] S.K.Prasad, J Rachna, O.I.Khalaf, D. N. Le, "Map Matching Algorithm: Real time location tracking for smart security Application", Telecommunication and Radio Engineering,79(13):1-14(2020)

[7]Homa Taghipour,Amir Parsa,Abolfazl Mohammadin,"A dynamic approach to predict travel time in real time using data driven techniques and comprehensive data sources", Elsevier(2020) Hashemi M., Karimi H.A., "A critical review of real-time map-matching algorithms: Current issues and future directions", Computers, Environment and Urban Systems 48, 153–165 (2014)

[8] Hu G., Shao J., Liu F., Wang Y., Shen H.T.,"If-matching: Towards accurate map-matching with information fusion", IEEE Transactions on Knowledge and Data Engineering 29(1), 114–127 (2017)

[9] Geoff Boeing, "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks", July 2017 Computers Environment and Urban Systems 65:126-139, DOI: 10.1016 /j.compenvurbsys. 2017.05.004

[10] Matthew Treinish, Ivan Carvalho, Georgios Tsilimis Gounakis, Nahum Sa, "networkx: A high-performance graph library for python", Attribution 4.0 International License (CC by 4.0)

BIOGRAPHIES

Prof. Smita Jangale is the Deputy Head of Information Technology Department at Vivekanand Education Society's Institute of Technology.

Rutuja Rajhans is currently pursuing her B.E. in Information Technology from Vivekanand Education Society's Institute of Technology.

Anuja Shetye is currently pursuing her B.E. in Information Technology from Vivekanand Education Society's Institute of Technology.

Ritik Wadhwani is currently pursuing his B.E. in Information Technology from Vivekanand Education Society's Institute of Technology.