8-20-2021

Forest Park Trail Monitoring

Adan Robles
Portland State University

Colton S. Maybee
Portland State University

Erin Dougherty
Portland State University

Forest Park Trail Monitoring

August 2021

First A, Adan Robles, *IEEE*

***Abstract*—Forest Park, one of the largest public parks in the United States with over 40 trails to pick from when planning a hiking trip. One of the main problems this park has is that there are too many trails, and a lot of the trails extend over 3 miles. Due to these circumstances' trails are not checked frequently and hikers are forced to hike trails in the area with no warnings of potential hazards they can encounter. In this paper I researched how Forest Park currently monitors its trails and then set up a goal to solve the problem. We plan on solving this problem using 3 different systems, an android-based app, a web front-end, and a python backend. In hopes to help with trail monitoring using robots/drones, we will discuss existing algorithms used to keep track of Forest Park trails and make recommendations on what the best path planning algorithm is.**

I. Introduction

Located in northwest Portland Oregon, Forest Park is one of the largest urban parks in the U.S, holding more than 80 miles of maintained trails, fire lanes and forest roads that are perfect for hikers, runners and those who want to explore nature. Forest Park attracts many individuals because it has nearly two dozen trails for every skill level to choose from when considering going on a hiking adventure. The park's 5,200 acres begs the question: how does this large urban park monitor its trails? The urban park is currently monitored by the Forest Park Conservancy's staff while also working closely with Portland Parks & Recreation and other volunteers year-round to ensure the safety and stability of the trails (forest park conservancy, 2021). Due to how large the urban forest is, it is difficult to keep track of all the trails in the forest. Volunteers do check these trails along the seasons, but they do it on pen and paper which makes it easier to lose data or not have it reported. Infrequent trail reports have helped us come up with an overall goal to solve this problem with trail monitoring. Our approach to solve this problem is to develop an android based mobile app that will allow users to report a trail when it has last been hiked, while also displaying an option to report hazards. We plan on hosting a run/walk for volunteers that simultaneously serves as data collection for the park and the collected data will be documented on the app and then transferred onto a public website where users can find all the data collected, signup information, educational material and much more. After gathering volunteer data, our overall end goal will be using robots/drones to gather all our trail data.

II. Goals for our system

A. App

Our end goal for the android-based mobile app is to have it be easy to navigate and make it accessible to the public so that anyone who goes hiking along the Forest Park trails can report when a trail has been hiked. This android app will have functionalities such as having the ability to pull GPS information to notify the user when he/she has walked part of the trail and automatically transfer that information to a map database so that when a trail has been checked it can be updated. Most importantly, the app will also have a functionality for reporting hazards along trails. Users will have a checklist of possible hazards to report at any given GPS location with the option to take a photo of the spotted hazard. More about our progress with the mobile app found on Colton Maybee's technical paper.

B. Web Front-End

The goal for the front-end part of the project is to have an online public website with all of Forest Park trails displayed. Some of the features the website will be responsible for are displaying current trail status, number of obstructions, obstruction type with a short description describing what the obstacle is, what time and day it was found etc. Trails will be colored arbitrarily based on how long they have last been hiked and will also display markers along trails when a hazard has been reported. These markers will be responsible for displaying a photo taken by the hiker who found the hazard. An entire web map of all Forest Park trails will be embedded onto our website to make all our trail data interactive to online users.

C. Python Back-End

The back end of this project will be where we store and manipulate algorithms within Forest Park trails. Some of the algorithms we will be using are Rapidly exploring Random Trees (RRTs), A* search algorithms, and other algorithms and their variants. We will be using these algorithms when we host our 5k event to keep track of the location of volunteers while also having the algorithm calculate shorter paths for them to take. Eventually our overall end goal with the Python back end is to have robots/drones use an algorithm to our liking and have them rapidly explore trails. We began running A* search algorithm within our trails and our progress with the Python back end can be found on Erin Dougherty's Jupiter notebook.

D. My part of the project

Throughout the NSF REU, I have overseen the web front-end portion of this project. I have specifically been working with the Folium Python library alongside the operating system (os) module to plot Forest Park trails on Python. Folium allows you to build interactive web maps and gives you the ability to manipulate your data in Python, which can then be visualized using the open-source JavaScript library. OS is a Python module that allows you to interact with the operating system, in this case creating paths between GPS latitude and longitude coordinates. Folium can generate an interactive web map by simply knowing the starting latitude and longitude coordinates of your map, while os.path processes files usually containing coordinate information to make a path between coordinates. The website geojson.io was used as a simple way to extract coordinate data from GPX files found on publicly available websites such as Gaiagps or Alltrails. Gaiagps is a publicly available website where you can find trail data for any trail in the world, but

after looking for Forest Park trails specifically, I found that Alltrails is an overall better alternative because Gaiagps does not provide trail data for all trails, which Alltrails does. The biggest challenges I faced while working on the web front-end was getting started initially since I had no experience with Python. Researching and watching tutorials on how other people have plotted trails on Python helped. My first attempt to plotting trails on Python, I used openrouteservice which is a library in Python that allows you to make a route in between coordinate points. Openrouteservice worked well, but the problem I had was that I could not figure out how to generate an API key that could plot all the coordinate points for the Maple and Wildwood trail loop on Forest Park. I overcame this challenge by importing the GPX file of the Maple and Wildwood Trail loop on Google Maps to visualize what the trail looked like and from there I followed two different tutorials and learned that geojson.io gave me all the coordinate information of the trail loop when I imported the GPX file. Copying and pasting the trail loop information onto a file and saving it as a .json file on Python then allowed the OS module to make a path between every coordinate on the file, successfully plotting the trail loop.

Once I was able to plot a Forest Park trail, I began working with folium markers. Folium has built in custom marker colors and icon types from Bootstrap and Fontawesome which can be applied to a web map using the folium.Marker plugin on Python. You can place a marker anywhere along a trail if you have the latitude and longitude coordinates of where you want the marker to be placed. After playing around with the built in Folium markers I began looking at making custom icons within markers. I was able to do so using the folium.features.CustomIcon plugin incase if we want to have our own custom markers. Also, I investigated how to display images once you click on markers. I found that IFrame and base64 libraries allow you to do so. From folium you can import IFrame which is an html that allows you to insert content from another source onto a web page, in this case a web map. IFrame allows you to display png and jpg images using IFrame and base64 libraries. Base64 is used to format and encode those images to be displayed onto a marker.

After being able to successfully display markers onto trails, I began coloring our trails. I found that geojson.io does not let you change the colors of a file imported, it only allows you to do so when you plot something manually, so I was unable to change the color of a trail. With the help from Colton Maybee, we were able to figure out how to change the color of trails, while also embedding links to every trail to show each individual trail once you click the link. Towards the end of the final week of the internship, I started working on our website. I was able to get the navigation part of the website going, all that's left is styling and adding information to our liking. Since the focus of the website was to have it display our trails, I was able to have a link going to display our Python web map of all the trails. In the website I used an IFrame which allows you to embed a website into a website and since our folium map is saved as an html, the website is able to display the map.

III. Graph based-backend

A. *Our Goal to cover all Forest Park Trails*

Our goal to cover all Forest Park trails will become more and more difficult as time passes by due to the season change and not having many volunteers or other hikers to report all the trails in the park. Another problem we will face is that we will not be able to rapidly explore the trails due to the amount of trails the urban forest has and how large the trails are. Our plan to solve these problems is to have drones report trail conditions. To do this we will have these drones run on an algorithm to rapidly explore a given area. Path planning or motion planning will play a very important role on this project once we start to implement existing algorithms into drones to monitor Forest Park trails.

B. *Algorithms that might fall short*

Some existing algorithms that might work are A* search, PRMs, and their multi-agent variants. A* search is a great algorithm that can be implemented on our drones, but the reason it might fall short is because it takes a path that covers the least number of grid cells to reach its destination (Akshay Kumar, et all., 2016). The A* algorithm works very similar to what Rapidly exploring Random Trees (RRTs) do, but instead they calculate the value of each node on the work area to get to an optimal ending point. Another reason the algorithm falls short is because it needs processors of high configuration to check various nodes successively (Akshay Kumar, et all., 2016). If the distance between the source and the goal is large, then the implementation of A* algorithm from the source and the goal involves the checking of too many adjacent nodes which will just increase processing time (Akshay Kumar, et all., 2016). However, the A* algorithms variant called the Time- Efficient A* algorithm might make this algorithm viable. The Time Efficient A* algorithm fetches all nodes but calculates the heuristics function's value only before the collision phase, reducing the processing time so that the robot can perform its work quickly (Akshay Kumar, et all., 2016). Figure 1 below explains in detail how the variant of the algorithm works.

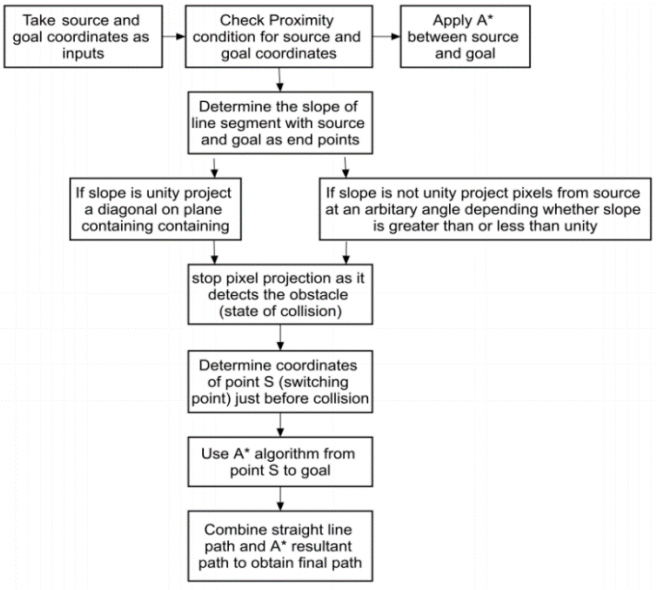

Figure 1: Flow chart of Time Efficient A* Algorithm (Akshay Kumar, et all., 2016)

Another algorithm that might work but can fall short is the Probabilistic Road Maps (PRMs) algorithm. This algorithm generates random samples in graph, defines them as vertices if they are collision-free, connects them with the nearby vertices if the connection is collision-free, and then searches to obtain the optimal path with appropriate searching algorithm (L. Ou.,2018), PRMs might be a viable option due to its probabilistic completeness and how easy it is to apply onto a complex environment due to not needing to characterize the space, but they do not work well for some

narrow passages (L. Ou.,2018), meaning that this algorithm can be a hit or miss to our project.

C. *Recommended existing algorithm*

The existing algorithm I recommend that can be used to rapidly explore all the trails in Forest Park is the RRT algorithm. Rapidly exploring Random Trees (RRTs) is an algorithm that is used to efficiently search high dimensional spaces by randomly building a space filling tree. The way RRTs work is, it creates a path between an initial point to an overall end goal. RRTs begin at their initial point, and then expand nodes randomly until the tree has reached its end goal, which most of the time ends up being the shortest path. Some of the advantages of RRTs for this project are; having a heavy bias towards unexplored places, consistent behavior due to the lead of distribution of vertices approaching sampling, probabilistically complete under very general conditions, facilitates performance analysis, always remains connected even though edges remain minimal, etc. (LaValle, Steven M., 1998). I recommend RRTs because we will have to have our robots/drones do path planning and RRTs and their variants seem to do well in dynamic environments.

IV. Conclusion

Path planning will undoubtedly be the focus of our project once we are able to get all our systems going. Of course, it will take lots of time and research, but once we are able to figure out a specific algorithm and implement it successfully to robots/drones then we should be able to keep track of the trails located in Forest Park. The app, web, and Python ends will serve as a guide to help visually understand the purpose of this paper and will help guide the next steps. Some recommendations I would like to make is to first do background research on the things we have already done so that it all makes sense. I would also like to mention that even though RRTs hold many advantages, there are other algorithms that might be better, start by looking at other algorithms and their variants and from there look for what will correlate best with our specific goal with Forest Park trails.

References

Forest Park Conservancy. 2021. *Maps | Forest Park Conservancy*. [online] Available at: <https://forestparkconservancy.org/forest-park/maps/>.

LaValle, Steven M. "Rapidly-exploring random trees: A new tool for path planning." (1998): 98-11.

Akshay Kumar Guruji, Himansh Agarwal, D.K.Parsediya, Time-efficient A* Algorithm for Robot Path Planning, Procedia Technology, Volume 23,2016,Pages 144-149, ISSN 2212-0173, https://doi.org/10.1016/j.protcy.2016.03.010. (https://www.sciencedirect.com/science/article/pii/S2212017316300111)

L. Ou, W. Liu, X. Yan, Y. Chen and J. Liang, "A Review of Representation, Model, Algorithm and Constraints for Mobile Robot Path Planning," 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), 2018, pp. 563-569, doi:10.1109/ITOEC.2018.8740620.

C. Maybee, Digitally reporting trail obstructions in Forest Park,“ 2021.