Finding entertainments for travellers

Team Members: Junchao Liu, Haoxun Ou

Instructor: Greg Baker

CMPT353

Summer 2020

Table of Content

Abstract:	**1**
Introduction:	**1**
Data:	**1**
Method:	**2**
Discussion:	**4**
Conclusion:	**5**
Project Experience Summary:	**5**
Reference:	**6**

Abstract:

In this project, we tried to find places of interest and provide suggestions to users about what activity they could enjoy when they were going to travel in some specific route. We used the OpenStreetMap(OSM), and Exif data that user input as references, calculating route that may be visited by the user, hence getting some useful output for them.

Introduction:

We all love to travel, but what is the first thing you will do before you travel? I guess most people will say do some research for places they wish to visit from the internet. However, most touring information found online do not connect well with one another. Therefore, users have to spend more time doing research planning for their travelling route. The program we had created can help users to plan their travels more efficiently. Firstly, users should collect images of the places they wish to visit the most, and their traveling plan. After that, we could provide them what other entertainment (sightseeing tour, biking, boat tours, kayaking, Etc.) along the path of the journey could be considered as options. Therefore, using our programs, users can discover more places to go and save time doing planning for their trip to a new city.

Data:

In this project, we mainly used the OSM data provided by the community. However, after inspecting the OSM data, it's not sufficient to answer our question.

The OSM contained mostly restaurants, parking, and benches, and it's not what we wanted to report to our users. Hence another problem for OSM was that the dataset itself contained too many different amenities, and they're no specific amenities for a place of interest. If we still want to get output from the OSM, it would require a large amount of hard-coding, and it was not sufficient when we wish to adapt this code to another place.

Although OSM was not perfect for reporting the result, it's an excellent tool for predicting route. To solve the output issue, we found another dataset from the local government contained specifically with a place of activities and attractions in Vancouver.

After joining these two dataset, we had an expended OSM that we could use to predict the route, to avoid ambiguity, we set an indicator variable to indicate which dataset the row came from, this also helped us get the final result easier. After all these data manipulation processes, we did not do any further data cleaning process.

The input files we would use were geotagged Exif files that had been put into ordered, reading these Exif files from users, we could get users' traveling plan. With the joined dataset and Exif files, we could start using these information and apply them to get the result.

Method:

As stated in our question, when the input was a set of ordered Exif files, we wanted to know the place of attractions/activities in that route. The first thing was to find the route that possibly travelled by the user.

We developed a simple pathfinding algorithm that could find a possible route in two-point, the idea was simple, and we would later use a trivial diagram to explain the process.

1. The first step was to consider the map as a two-dimension plot, longitude in the x-axis, latitude in the y-axis so that the x-axis will range from $longitude.min()$ to $longitude.max()$.
2. The second step was to divide the map into even size squares.
3. After the map had divided into small squares, we could count each square's number of amenity for further use.
4. We started finding the route between two points. From here we would introduce our simple map example first; figure 1 showed a map with the x-axis which has been divide into two bins. We would use this figure to illustrate three cases we would encounter when doing the route finding.

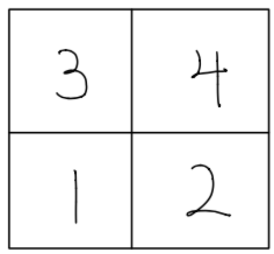

Figure 1: Simple Map

4.(1) The start was in the same horizontal line as End: We would move horizontally from Start to End and record every step we travelled. For example, if $Start = 1$ and $End = 2$, the route we observed was simply 1——2.

4.(2)The start was in the same vertical line as End: We would move vertically from Start to End and record every step we travelled. Refer back to Figure 1; this was the case the $Start = 1$, $End = 3$, and the route would be 1 ——3.

4.(3) This is the most complicated case; when we had Start and End be in different rows and columns, we would need the count of amenities in each square to help us make a decision. For example, if $Start = 1$ and $End = 4$, in the first step, we could either go up to area 3 or travel right to area 2. Now we would compare the count of amenity in area 3 and area 2. If the number of amenity in area 3 is greater than the number of amenity in area 2, then we would go up to area 3 and decide which way to go next. For instance, the number of amenity in area 3 was 25 and the number of amenity in area 2 was 10, 25 > 10 so we would choose to go to area 3. Now $Start = 3$, $End = 4$ and we were back to case 4.(1). Finally, if $Start = 1$ and $End = 4$, the route will be 1——3——4.

5. After getting the route from two points, we could easily apply it recursively and get route from multiple locations.
6. With the route calculated, we could then get the information needed to answer our question and output to users.

Here are a few lines of results for the input Exif files from SFU—Gastown—Canada Place—Stanley Park.

name	amenity	Keywords
Aisin-Gioro Fine Art Gallery Ltd	Art Galleries	
Dr. Sun Yat-Sen Classical Chinese Garden	Gardens	Museums ; Historic & Heritage Sites ; Science &
Pacific Coach	Transportation - Bus	Tours - Sightseeing Tours ;
Science World at Telus World of Science	Science & Nature	Museums ;
Sins of the City Walking Tours	Walking Tours	Museums ; Historic & Heritage Sites ; Sightseeing

Table 2: sample results

Discussion:

Although the simple route-finding method was applicable to our project, it still had some limitations that we might get the wrong output(route) from it.

1. When the Start point and End point were in the same vertical or horizontal line, our program would go directly from Start to End and get the route between them. Which could create a problem when there's something we could not easily cross in the middle. For example, if our $Start = SFU$ and $End = DeepCove$(in North Vancouver), our program would predict it go directly North up from SFU to Deep Cove, which was impossible because there's no road in the middle.
2. We had free variable bins, which specify how many bins we want to divide our map into. It's hard to adjust the bins to the "right size", if bins were too large, then we included too little places and it's too small, we would include too little places. This adjustment was especially hard to make if we were not familiar with that place.

Conclusion:

The difficulty of this topic was to create the appropriate pathfinding algorithm, which we spent most of our time on. We tried a few approaches and finally came up with this simple pathfinding algorithm. It's not perfect, but it's applicable, we could foresee that if we had the data for all the rivers and traffic volume for the city, we would make a much better prediction of the route.

Project Experience Summary:

Haoxun Ou:

Finding entertainment options for travellers August 2020

- In a group of 2 to filter out location coordinates in geotagged photos by using python and conducted a project report.

Junchao Liu:

Finding entertainment options for travellers August 2020

- Purposed structures, timeline for the project.
- Gathered useful dataset and cleaned for usage.
- Designed a simple path finding algorithm.
- Conducted a formal report including all details and constraints.

Reference:

https://wiki.openstreetmap.org/wiki/Downloading_data

https://catalogue.data.gov.bc.ca/dataset/hellobc-activities-and-attractions-listing

https://www.quora.com/How-can-I-read-multiple-images-in-Python-presented-in-a-folder

https://developer.here.com/blog/getting-started-with-geocoding-exif-image-metadata-in-python3