Engineering: Summer in NUS

ZHUO ZHOU
Department of Industrial Engineering, College of Engineering, University of Illinois at Urbana-Champaign

INTRODUCTION

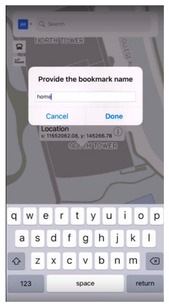
Like the places with highly intensive information like hospital and university, people who have little access to information would feel of living in the jungle: no direction, not knowing which way to reach the destination. Our objective is designing and developing a prototype Mobile App for a specific use case such as finding a certain place, meeting a certain professor in the university, going to the specific class, or even going to meet for a job interview.

Basic View of the App

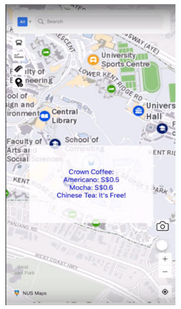
PROPOSED FEATURES ADDED

Saving Address

The runtime SDK we used for this feature is AGSBookmark.To implement the function, the users need to tap on anywhere on the screen and type a bookmark for the place that he or she wants to save. The screenshot of this step is as follows.



Create a Bookmark

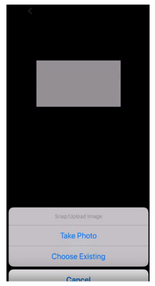


Promotion Pop-up

Location-based Promotional Pop-ups

We basically use sensing around the i4.0 building to detect the users current location in the form of latitude and longitude and using a function to convert the distance of the user and the specific store to meters. Once the distance is within a threshold, the promotion will be shown in the screen. The promotion pop-up will be shown as follows.

Sign Recognition and Location Input with Optical Character Recognition



Imagine a new student to the university or a student who wants to know the information of a specific office or room, such as the name of the professor in the office and the reservation availability of the study room, a way to easily get the information of the room in this case is particularly useful and saving time.

To receive the information of the room, what we use is a machine learning technique. We basically allow the access to camera on the app to take a picture for the name tag of the room. Once the engine detect the name in the correct way, the information will be shown on the screen.

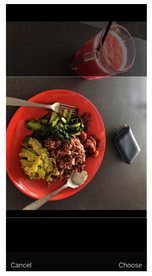
The machine learning we implement is Tesseract, which is an optical character recognition engine and it can be implemented on various operating system. The text recognition feature is on another view.

Limitaion:

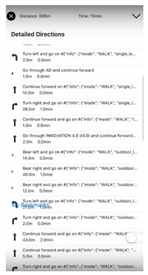
We do have limitation on this feature: The trained data in OCR cannot always recognize the correct texts we want. We have to make sure that the image we choose to be detected is in the best quality for Tesseract Engine to recognize. The ways to improve the quality of image for Tesseract are Rescaling, Binarization, Noise remove and so on. Another way to denoise and reduce the error is autocorrection in IOS.

Photo-based Location Input

Since we can use GPS, the location of the photo we took will be store on our phone. Users can select the image from the photo gallery on the phone through a button. By this way, users will get a route from the location of where they are right now to the location that they took the pictures. Here is an example of choosing existing photo.



Choosing Photo



Getting Routes

3D Indoor View

We also have a 3D Augmented Reality view as on the left side. Users can switch from 2D map view to 3D scene view in the App. This view is created for NUS students and staff enable room navigation it is differentiate itself from other map application and get users to the exact room they want to go.

Side Project: CVRP Model in NUS

Objective:

Our objective is to build a solver to compute a route for each vehicle to minimize the total time of transporting goods from a warehouse to random number of stores within Singapore.

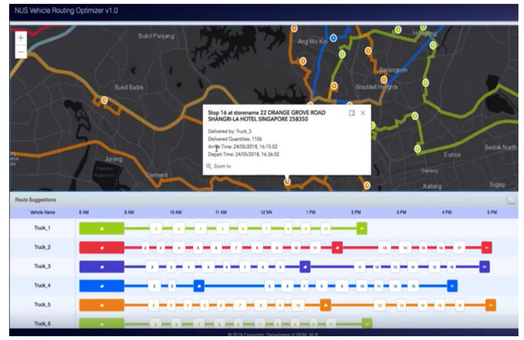
Constraints:

Each vehicle has its own capacity. We are given fix data for the convenience of our optimization. In the real case, the data including the number of stores, the location of the stores, the capacity of the stores and the time travel from store i to store j will be received from the server. Our server will receive those data to do real time calculation.

Solver:

The Primary solver we need is the local-guided search in Google-or tool. (https://developers.google.com/optimization/)
Below is a sample printed solution of our solver for the route of vehicle:
Route for vehicle 0:
0 Load(0) -> 56 Load(1804) -> 80 Load(3626) -> 111 Load(5446) -> 141 Load(7370) -> 157 Load(9192) -> 110 Load(10688) -> 133 Load(12019) -> 30 Load (13515) -> 0 Load(13515)
Time of the route: 300min
Load of the route: 13515
Distance of the route: 53927.520000000004
Finishing time: 2019-06-24 13:00:00
Time Utilisation: 55.6%

After we connected with our front-end developer, we got our sample visualized result:



Our expectation for the solver is to not only minimize thetotal time, but also minimize the difference of time utilisation among each vehicle as no driver wants to work too longer than others.

ACKNOWLEDGEMENTS

Firstly, I would like to convey my special thanks to my professor(Prof. Andrew Lim) as our principal , my project manager(Raymond Huang), who gave us valuable ideas about our project(Experiential Wayfinding) and Bilguun Batbold for the technical advice especially about the code.
Secondly, I would like to thank to the other members in the team including Sunny, JiaXin, Roseele and RuiXue who also help us solve our problems during the project.
Finally, I would like to thank to my teammates Keying Chen and Jasper Huang who gave me lots of helps and encourage me.

ILLINOIS