Collaborable Indoor Positioning Project

Guanting Liu, guanting.liu@sv.cmu.edu, GitHub link

Acknowledgement: Dr. Ian Lane

1. Introduction

CMU-SV-Indoor-Swift is a collaborable indoor positioning project that aims at providing indoor positioning service specifically to the community at Carnegie Mellon University's Silicon Valley Campus. Written in Swift programming language, it is developed specifically for iOS devices, and can be a reference when building similar apps for Android devices.

2. App Design Overview

CMU-SV-Indoor-Swift is first itself a runnable app that provides indoor and outdoor positioning function.

The function of this app is divided in to three modules:

1. **Indoor Positioning.** The indoor positioning feature is implemented with IndoorAtlas API, which uses the magnetometer on the iOS device to sense the indoor magnetic field and match the data to an indoor position where the magnetic field has been pre-recorded.
2. **Outdoor Positioning**. The outdoor positioning feature is provide by Apple's CoreLocation framework which uses the GPS and cellular chips on the iOS device to triangulate conventional position which is accurate enough when used outdoor.
3. **Position Visualization.** The position visualization feature is implemented with the widely used Google Maps API. The indoor maps, a.k.a. floorplan images, are implemented as GMSGroudOverlay objects that are drawn on top of the Google map.

Both indoor (when available) and outdoor positions are provided as geographical coordinates to the Google map view in real time, which then updates the position marker indicating the user's position.

Fig.1 illustrates the role of these three modules

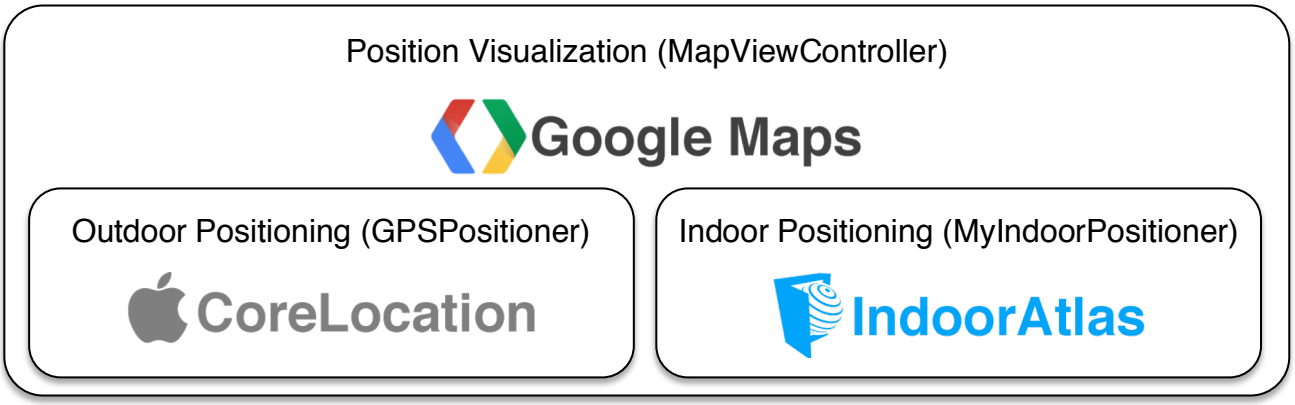

Fig.1 The three modules of the app

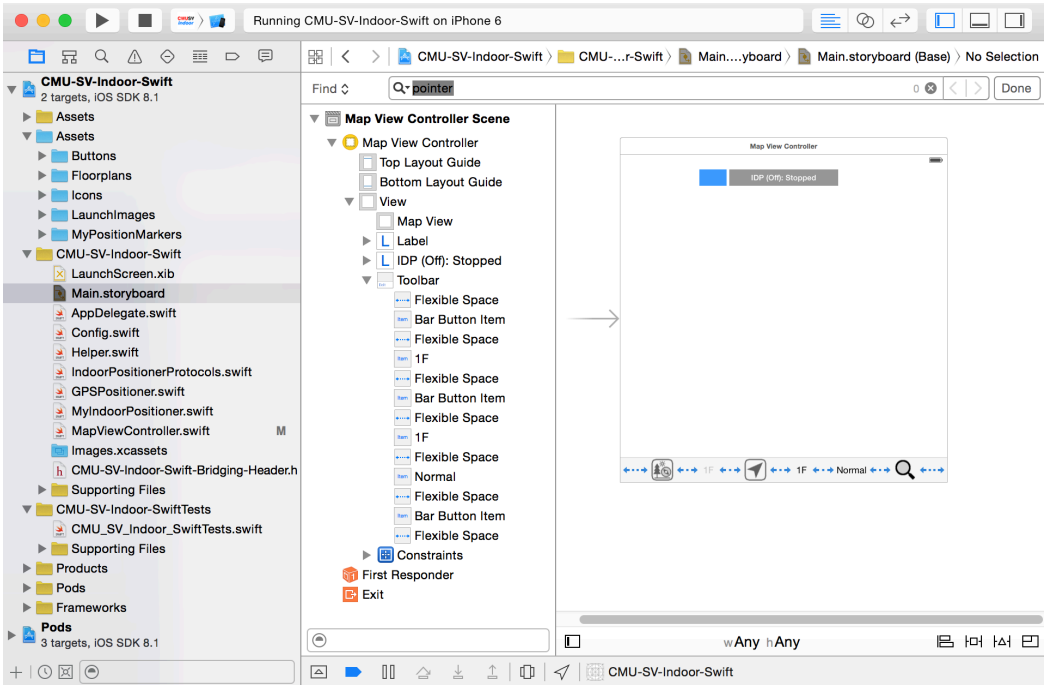

Fig.2 Project files

Fig.2 shows an over view of the project files:

- The **Assets** folder stores user interface icons and floorplan images.
- **Main.storyboard** defines the user interface of the app.
- Three main classes - **GPSPositioner**, **MyIndoorPositioner**, **MapViewController**, have been defined to implement the functions of the three modules accordingly.

3. User Interface, Operating Principles And Performance

3.1 User interface

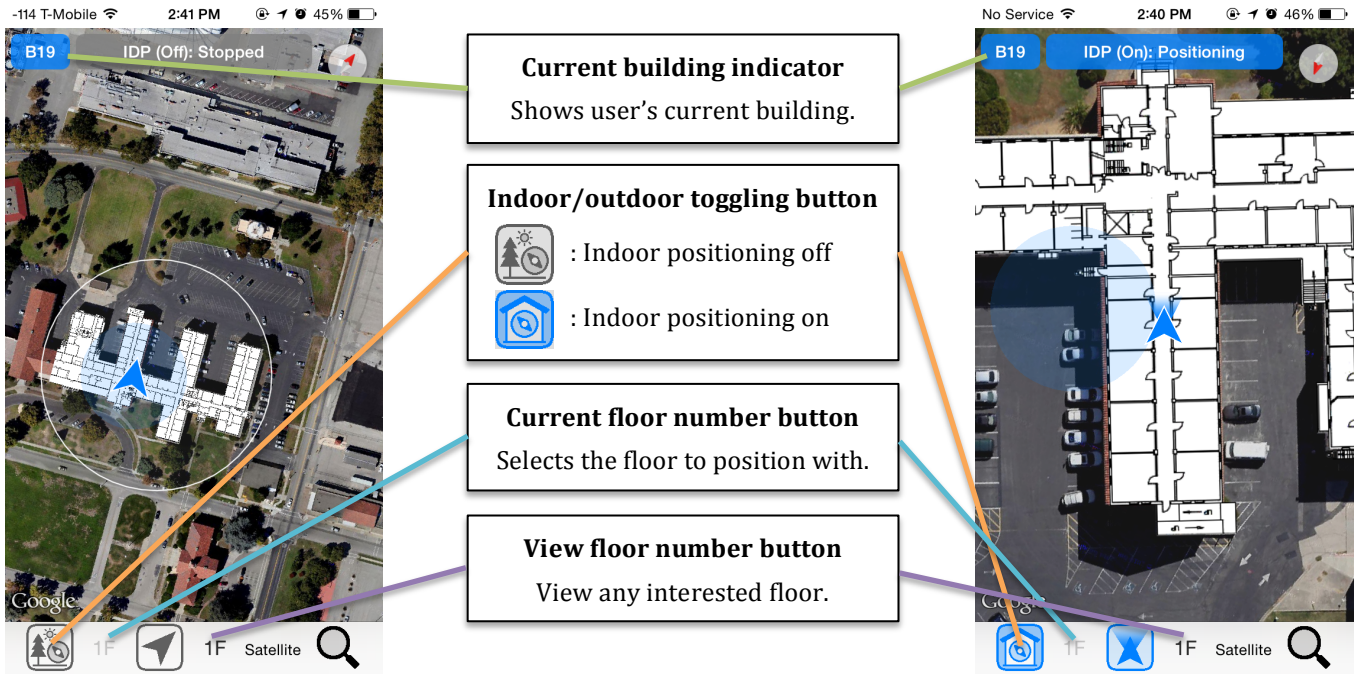

Fig.3 User interface of CMU-SV-Indoor-Swift

As shown in Fig.3, the app has a single screen containing the Google map and some UI labels and buttons.

3.2 Determining current building and floor

Currently the user still has to turn on or off the indoor positioning module manually according to the context. When indoor positioning is on and functioning, the position marker locates to the coordinate return by IndoorAtlas. When indoor positioning is off, the position marker locates to the center coordinate of the GPS position object, which typically has a horizontal accuracy radius. The horizontal accuracy radius could be quite significant when indoor, due to distorted indoor GPS signal.

Additionally, when positioning indoor, the user has to manually choose which floor of the current building he or she is currently on. This is due to the current limitation of the IndoorAtlas API, the inability to differentiate floors that overlap together vertically.

However, as a convenience, the app automatically determines the current building for the users. This is done by defining a range circle, which is essentially a center coordinate and a range radius in meters, for each building. The app then uses the real time GPS position object along with the range circles of all the buildings to determine if the user is "within" a building and which specific building. As long as the GPS position centers within the range circle of a building, the user is considered 'within' that building, even if he or she is in fact outside that building; otherwise, the user is consider within no building. This simple and primitive algorithm works well given the situation where buildings (Building 19 and Building 23) are distinctively separated from each other and the user has to manually turn on indoor positioning. This saves the user from needing to select the building when using indoor positioning.

However, due to the inaccurate nature of GPS signals received indoor, occasionally there could be misjudgments. A typical scenario is that when the user is indoor, the GPS signal could shift outside that building for a short moment. It is important to define an appropriate range radius for a building and/or consider defining minimum time threshold to trigger a building change event.

In future improvements, a small modification can be made to this algorithm such that it checks if the GPS position center is within the horizontal projection of any building rather than a range circle, i.e. if a geographic point is inside a geographic polygon. This would cost some more CPU cycles and computing power. However, the algorithm still could not precisely differentiate indoor and outdoor, and even the aforementioned misjudgments could still occur due to inaccurate GPS signals received indoor.

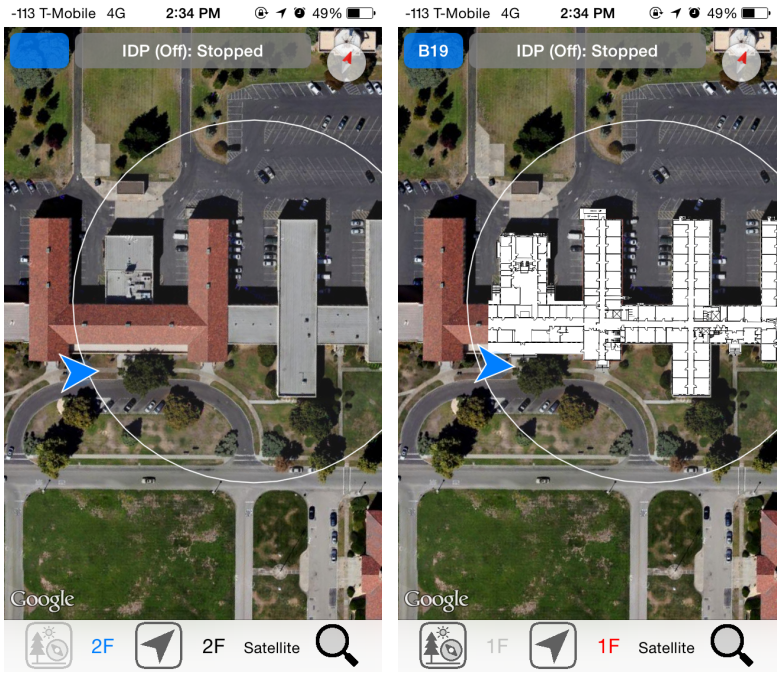

Fig.4 Range circle senses user's current building

As shown in Fig.4, the range circle of Building 19 is highlighted with white stroke. Before the user walked into the circle, he was considered within no building and indoor positioning on/off switch was disabled. As soon as he walked into that circle, he was considered within Building 19, and the indoor positioning on/off switch was enabled, but not turned on automatically. When he enters the building thereafter and needs indoor positioning, he can simply turn it on and select the correct floor number, without needing to select the current building.

3.3 Indoor positioning with IndoorAtlas API

Before IndoorAtlas can determine indoor positions based on the indoor magnetic field, it needs to learn about the magnetic field characteristics in those positions first. Magnetic field data must be collected and uploaded to IndoorAtlas cloud server. Considering the whole API as a machine-learning algorithm, this is essentially the training phase. And it involves several steps.

(1) Upload all the floorplan images to IndoorAtlas and align them properly.

In this project, the floorplan images uploaded are only used as the reference when collecting magnetic field data. And they must be the same size and scale as those floorplan images overlaid on top of the Google map. Notice that these floorplan images are not be visually aligned with the map provided by IndoorAtlas, which is not a Google map. At the end of the day, a coordinate provided by IndoorAtlas would position the user's marker on a corresponding floorplan image above Google map. It is critical to align the same floorplan image on IndoorAtlas' map and on the app's Google map so that the same pixel on the two floorplan image copies have coordinates as close as possible. The way the floorplan image is aligned in these two places are different. On IndoorAtlas, it is aligned by three pixel-to-coordinate mappings. Where as on Google map, it is positioned by specifying three parameters - the center coordinate, the scale and the bearing.

To help the two images copies align with each other, several markers are drawn at the corners of a floorplan image. The floorplan image is first aligned on the Google map of the app. Then run the app on the simulator. With the help of a method called **mapView(mapView:, didTapAtCoordinate coordinate:)**, which is a handy

Google map API method that is not really used when using the app, when the center of a marker is tapped by the mouse pointer, the corresponding coordinate would be printed. Using this method, the coordinates of all the three markers can be known, and the floorplan image can be properly aligned in the IndoorAtlas account.

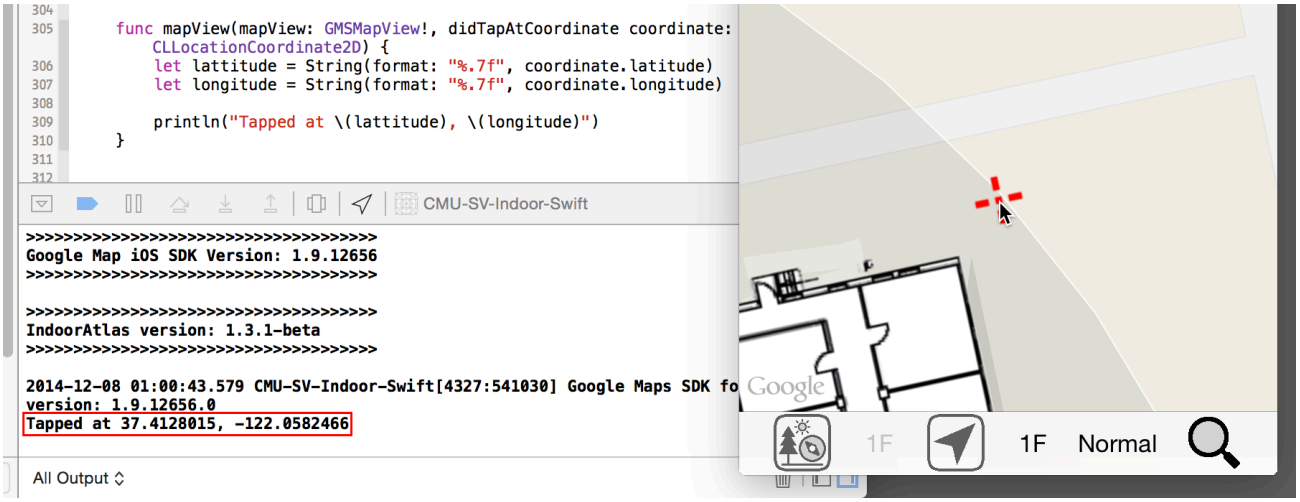

Fig.5 Reading the coordinate of a floorplan image alignment marker

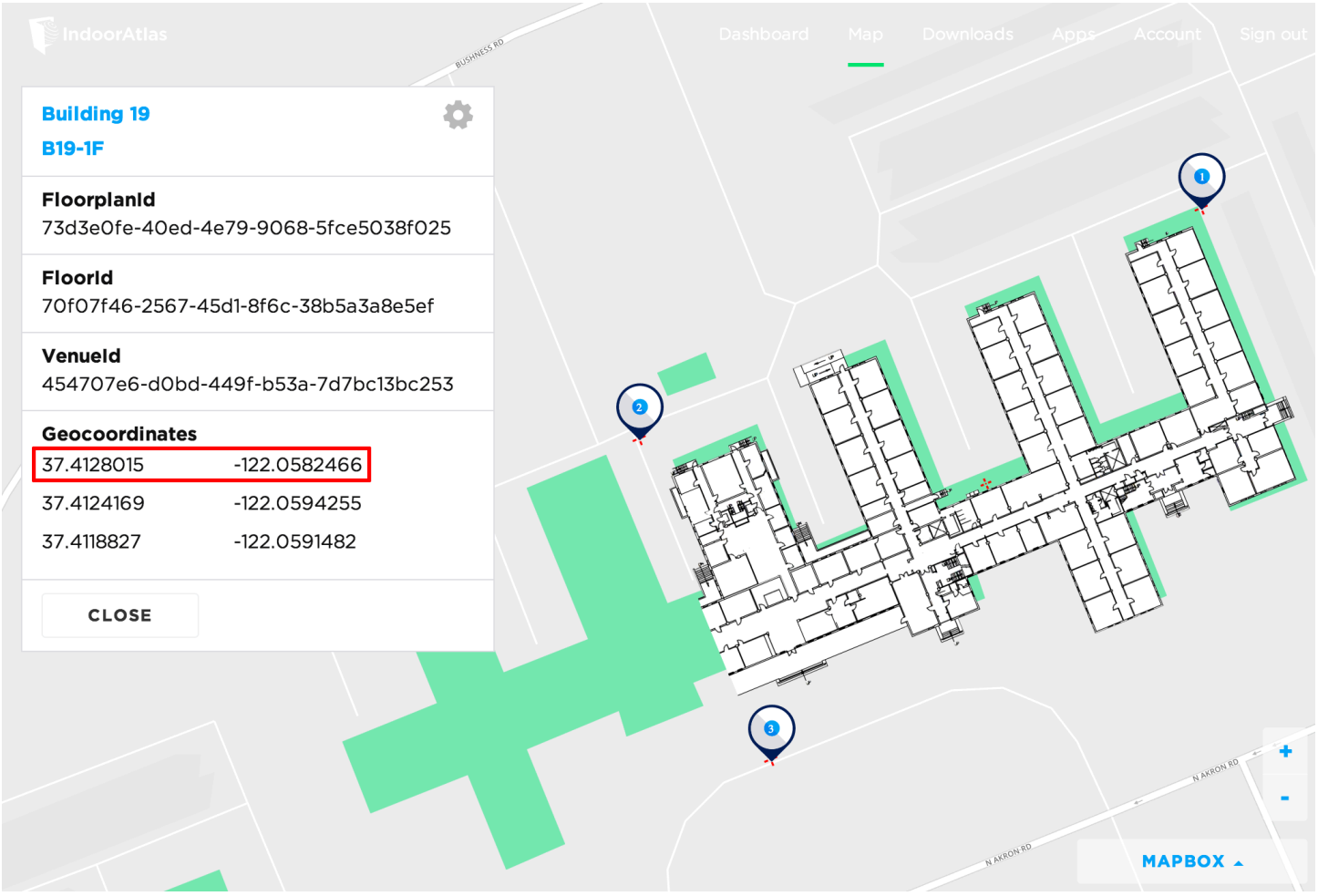

Fig.6 An aligned floorplan image on IndoorAtlas Server.

(2) Collect magnetic field data of each floorplan and upload the data.

Using an iOS or Android device with magnetometer, run the official IndoorAtlas app and open an uploaded

floorplan. Draw a small route segment on the floorplan, it can be a straight line or a curve. Record the magnetic field distribution of the route by walking through the route at a constant and appropriate speed. Validate the route by recording the magnetic field of the same route again in 'validation mode', in either the same or the opposite direction. Repeat until enough routes are defined to cover all the interested indoor positions, and the magnetic field data of each route is both collected and validated. Upload all the recorded route segments to IndoorAtlas server. All the data would be private to the IndoorAtlas account used. Note that at any moment, any recorded route can be deleted or recorded again, even after the route has been uploaded to the server previously. It is a good idea to define short route segments to allow possibly smaller modification in the future.

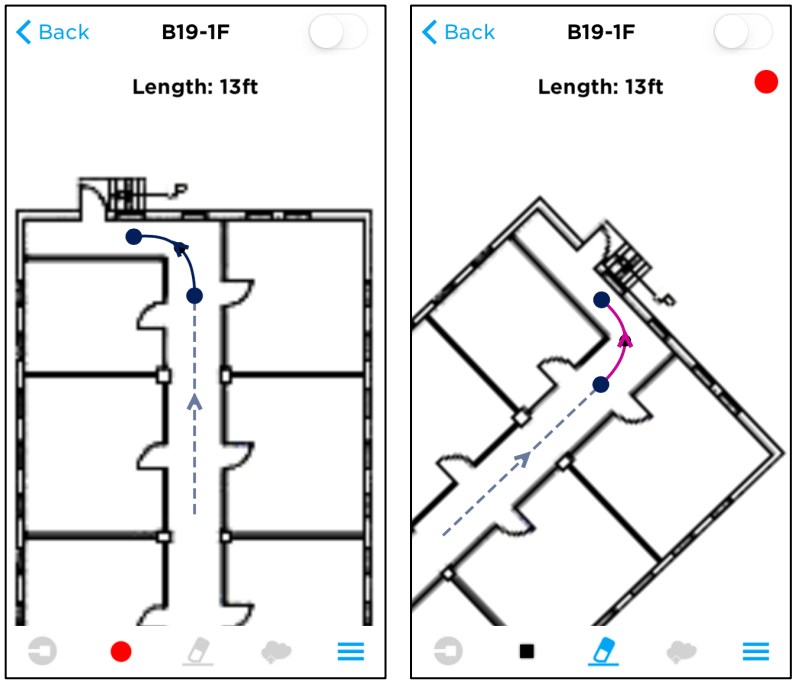

Fig.7 Example of drawing a route and recording the magnetic field.

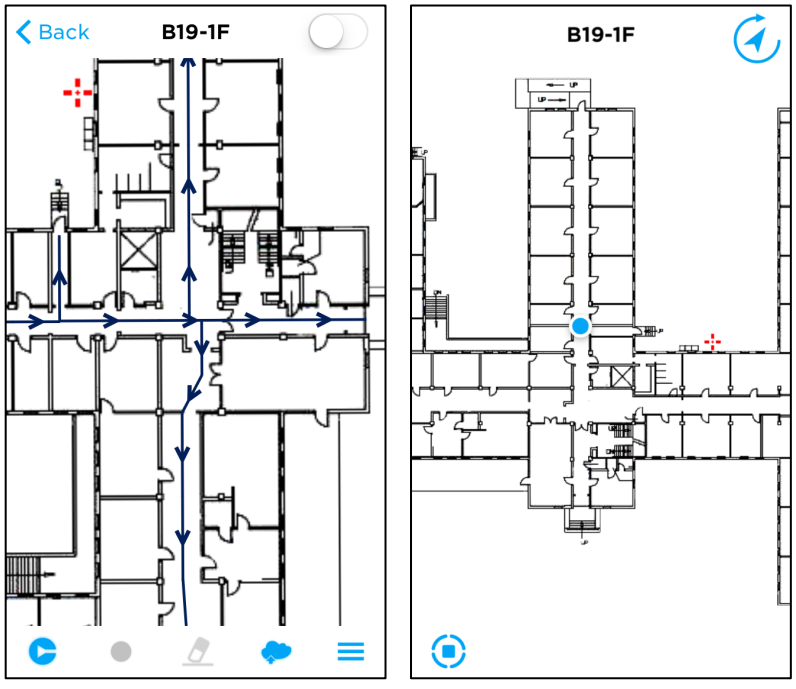

Fig.8 Viewing the recorded routes and testing indoor positioning in the IndoorAtlas official app

After IndoorAtlas has learned about the magnetic field of a floorplan, the app can simply call the API to query the indoor position within that floorplan, following the IndoorAtlas SDK documentation.

3.4 Indoor positioning accuracy and device adaptability

The indoor positioning accuracy of IndoorAtlas can be discussed in two directions – the axial direction along the route and the perpendicular direction to the route.

As soon as the app starts indoor positioning, it typically needs several seconds and/or a several-step movement of the user before the indoor position can be provided or gets fixed and stabilized. Once stabilized, the error in the axial direction along the route is typically less than two meters, which implies an accuracy good enough for most indoor location based services.

The accuracy in the direction perpendicular to the route is determined by several factors: 1. The width of the space perpendicular to the route, the degree of variation of the magnetic field across the route, the density of the parallelly defined routes in the same walkway or room. For most narrow indoor walkways, typically only one route is defined along the center axial, and hence the perpendicular accuracy is half the width of the walkway. For wider walkways or rooms, there are typically multiple routes forming a complex pattern and the perpendicular accuracy or even the axial accuracy would be different from place to place.

As a matter of fact, due to the nature of the way IndoorAtlas works, the axial and perpendicular accuracies would affect each other and an optimizing trade-off should be determined when defining the routes.

According to the IndoorAtlas' documentation, their API explicitly supports certain iOS devices and some Android models, and the magnetic field data is adaptable to any supported devices, even if the data submitted to the server had been provided by a device from the other platform. This app has been tested with an iPhone5 and an iPad3. Both devices can position indoor in an expected manner. All the magnetic field data has been collected and submitted with the iPhone5. The assumptions is that any iOS device that is running iOS 8.0 and above, is equipped with a magnetometer and has internet connection, should be able to use this app.

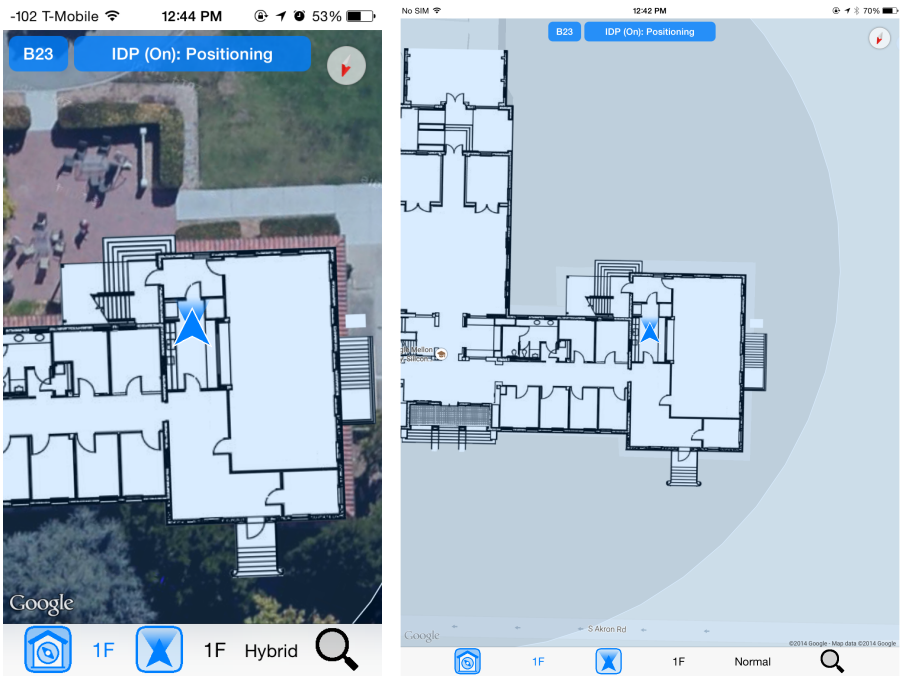

Fig.9 iPhone5 and iPad3 at kitchen on first floor at Building 23

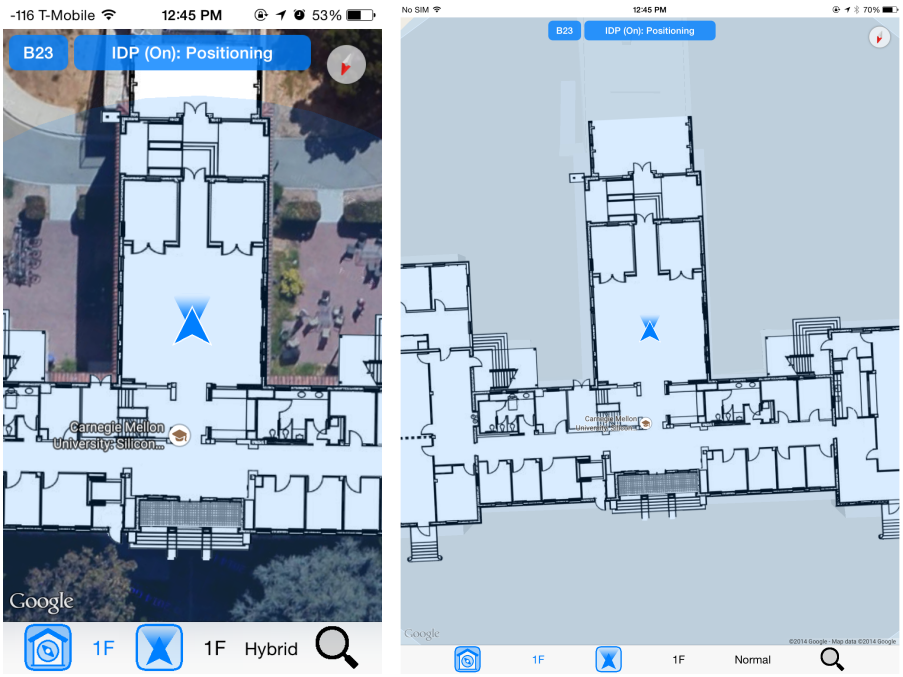

Fig.10 iPhone5 and iPad3 at student lounge on first floor at Building 23

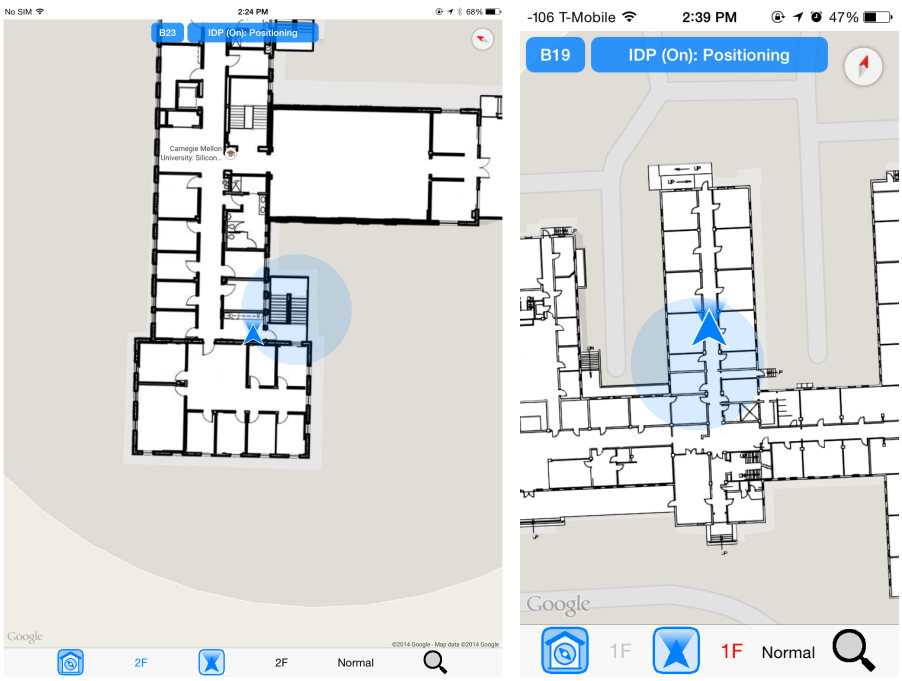

Fig.11 iPad at kitchen on second floor at Building 23,
iPhone at hallway on first floor at Building 19

4. Future Work and Collaborability

CMU-SV-Indoor-Swift is not only an indoor positioning app, but also a highly collaborable indoor positioning project in which people can collaborate in several different ways:

(1) Contribute to the improvement of the source code and the development of new features. The source code needs to be improved for better performance and maintained to adapt to future iOS and framework updates. Additionally, many useful features can be added to the app in the future, such as location searching and

indoor navigation. Moreover, as the app evolves, it would probably need to adopt the server-client pattern. All the publicly shared data, including the floorplan images and location specific information, would all be stored in one place - the server, in a database or as JSON, text and image files. Finally, it may be useful to transplant the app to the Android platform one day in the future.

(2) Indoor positioning technology researchers can take advantage of this project to speed up their research of different indoor positioning algorithm or applications. Using the already implemented basic indoor and outdoor position visualization feature of this project to test and validate new algorithm is very easy. As for now, all that needs to be done is to design a customized indoor positioner class that conforms to the predefined **IndoorPositioner** protocol and implement the required methods, make it a member property of the MapViewController and set MapViewController as its delegate when initialized. Finally the customized indoor positioner needs to call the methods defined in the **IndoorPositionerDelegate** protocol from the delegate at appropriate moments during the indoor positioning workflow. In fact, the MyIndoorPositioner class defined in this app, which provides reasonable indoor positioning performance, is an example of a customized indoor positioner.

(3) All people, technical or none technical, can contribute to this project by collecting magnetic field data with their iOS or Android devices. Firstly, some the magnetic field data submitted may be inaccurate or become outdated overtime. Secondly, every time the floorplan image is replaced with a newer one, the whole floorplan data needs to be recollected from zero. By engaging more people, adding or recollecting magnetic field data can be done easier.

5. Summary

This paper has presented a collaborable indoor positioning project, which not only emphasizes on real world application, but also aims to speed up the development and application of various indoor positioning technologies and engage more people into the world made of these technologies.