A Smartphone Based Indoor Navigation System

Shivam Verma*, Rohit Omanwar*, Sreejith V*, Meera GS†
*Dept. of Computer Science and Engineering, BITS Pilani - K K Birla Goa campus, Goa, India
{f2009680, h2012060, srev}@goa.bits-pilani.ac.in
†Thirdeye Networks, Trivandrum, India
meera.nanthi@gmail.com

***Abstract*—Although various solutions have been suggested for indoor navigation systems, most methods require the support of external physical hardware infrastructure. Due to the increase in complexity and cost of set-up of supporting hardware requirements, scalability will always be an issue with such systems. In this paper, we present the design of a smartphone based indoor navigation system. The proposed method uses on-device sensors for Dead-reckoning and is supported by a web based architecture, for easily creating indoor maps and providing an indoor location's information for navigation and localization. The system has been implemented and tested, and the results indicate that the approach is useful for navigation in indoor environments.**

***Index Terms*—Map Generation, Indoor Positioning, Indoor Navigation.**

I. Introduction

Existing navigation systems can be broadly classified into two major categories, *indoor* and *outdoor*. Most outdoor navigation techniques use satellite based navigation systems such as GPS, GLONASS, etc. to locate an object in any outdoor area. Such techniques work well in open spaces with a clear line of sight to the satellites, but may not perform well in an indoor environment, as the signals get scattered and attenuated by physical objects.

Over the past few decades, indoor navigation systems have been a very popular subject of research. A Radio Frequency(RF) based localization technique is proposed in [1]. For locating an object, this RF based approach examines the received signal strength(RSS) at multiple reference locations. However, such a system needs extensive hardware infrastructure support and is not easily extensible. The wide availability of mobile devices with multiple communication protocol support such as WiFi, Bluetooth, ZigBee, NFC etc. has given way to novel indoor navigation techniques. Wireless fingerprinting [2] [3] becomes more feasible due to the ubiquity of WiFi chips. However, the approach suffers from issues such as dependency on external signal distribution, fluctuation in RSS values, effect of human bodies on signal [4] etc. Other Bluetooth based techniques [5] utilize RSS values from BLE beacons placed in the user's environment and hence, suffer from similar issues.

Challenges in developing any indoor localization and navigation system include map generation, indoor localization, software development for the client platform, etc [6]. In this paper we present the design of an end-to-end solution which allows for map-generation, indoor localization and navigation with the help of an off-the-shelf smartphone. The proposed method is supported by a web based interface for users to easily create a map of any indoor location by capturing panoramic images. A smartphone application can then request for this map data from the web-server to localize and navigate the user to a destination via the shortest route. Visual feedback is provided to the user as he moves around in the area.

The paper is organized as follows; Section II explains the proposed architecture followed by the implementation details in III. Section IV concludes the paper along with the future work.

II. Architecture

Our proposed solution consists of two phases. The *Map Generation phase* is a one time process and is performed first for each indoor location. The *Localization and Navigation phase* will use the map generated in the first phase for user navigation.

A. Map Generation

Generating maps for any indoor location is a two-step process. In the first step, a user, the map-creator, captures multiple 360° panoramic images of the indoor environment. The user must ensure that these images are captured from each intersection of the pathways of the given indoor environment. We call these intersections, Points of Interest(PoIs). Once these images are captured, the user can generate a map by uploading the images to our online web based editor, the Map-Maker. The Map-Maker allows us to create and maintain maps for any number of indoor locations. Each location in the Map-Maker lists the panoramas captured by the user as shown in Figure 1. The user can then use the Map-Maker to define tags in each of these images. A tag is a quadrilateral area defined in the panoramic image as shown in Figure 2. Each of these tags represent a path that connect any two PoIs in the indoor location. The user then links this tag to another panorama representing an indoor path between these two PoIs. The Map-Maker also requests the user to enter a distance value when adding a new link. We provide this distance as the number of steps taken to reach the second PoI from the first. Additionally, the user must also mark an approximate location of magnetic north in each of these panoramic images. The process is same as adding tags to images and can be seen in Figure 2. This is done to determine the orientation of PoIs relative to each other while generating the indoor map.

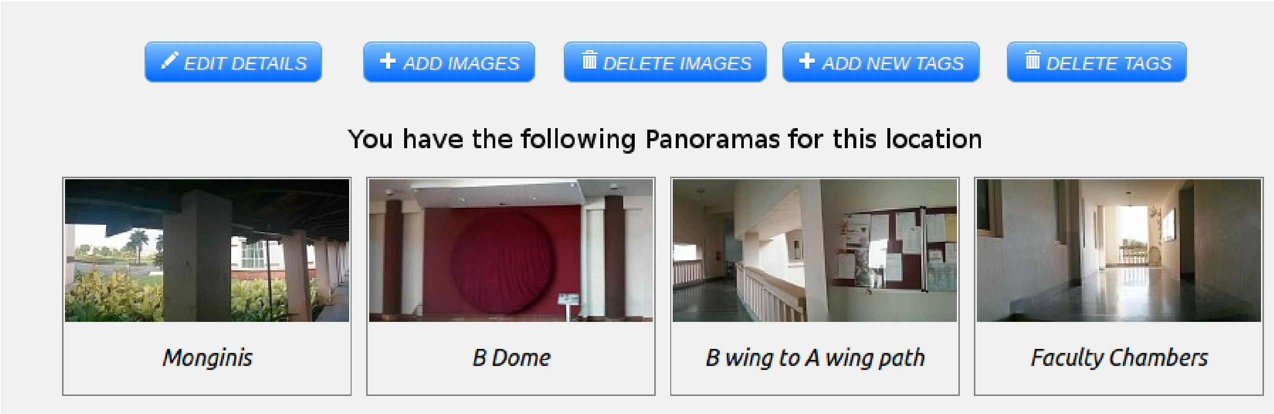

Fig. 1. Editing an Indoor Location in Map-Maker

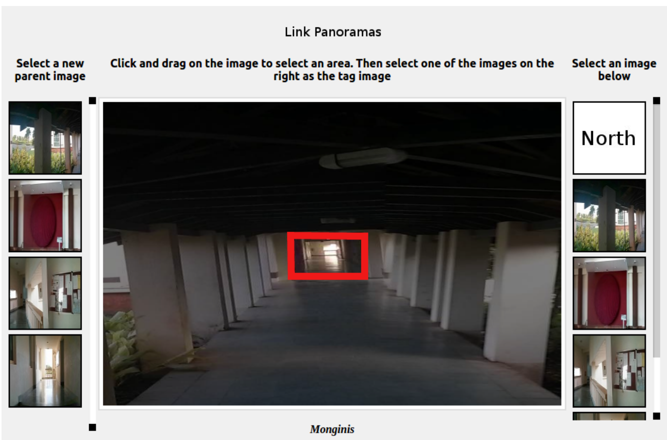

Fig. 2. Adding tags to panoramic images and linking images

Once these images are captured and updated, the web server will contain the list of PoIs, their relative orientation and their relationship with its neighboring PoIs. Hence, any indoor location can be mathematically represented as a graph G, where the points of intersection of indoor pathways are represented as nodes in the graph and the pathways connecting these nodes are represented as edges in G.

$G = (V, E)$;
$V = P_0, P_1, ...P_n$ where $P_i = i_{th}$ PoI;
$E = E_{0,0}, ...E_{n,n}$ where $E_{i,j}$ represents the edge between i_{th} and jth PoI and $1 \leq i \leq n$, $1 \leq j \leq n$, $i \leq j$;
$E_{ij} = \{w_{ij}, \theta_{ij}\}$ where w_{ij} = distance between i_{th} and j_{th} node, θ_{ij} is the angle of the edge with respect to the magnetic north direction;

When a user selects an indoor location in our proposed navigation system, we request this map data from the web-server to generate a visual representation of the indoor area. A generated indoor map can be seen in Figure 5. This map is then used for localization and navigation, as described next.

B. Localization and Navigation

The navigation module runs as a separate application. It enables the user to view a complete map of the area or navigate to a destination. When the user enters a destination, the directions via the shortest route are displayed to the user in text. These directions are also overlaid graphically on the map as shown in Figure 5. The application also displays the estimated location of the user on the map as the user moves around in the indoor environment.

To estimate the current location of the user, from a known initial position, we use Dead-reckoning algorithm. Dead-reckoning is the process of calculating the current position, with the help of previously calculated positions and thereby advancing that position based upon known or estimated data. The implementation of this technique can be done with the help of sensors available in our smartphones. In our implementation, we use the accelerometer and magnetometer sensor data to estimate the user's current location. Our approach of Dead-reckoning using these sensors is explained below.

1) Defining the Coordinate-Axes: The smartphone sensor framework on the Android OS uses a standard 3-axis coordinate system. For most sensors, the coordinate system is defined relative to the device's screen, when the device is held in its default orientation. The X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points outside from the screen face. In this system, the coordinates behind the screen have negative Z values. The axes are not swapped when the device's screen orientation changes, i.e., the sensor's coordinate system would never change as the device moves.

2) Sensor Data: At any given time, the magnetometer can be used to retrieve the orientation of the smartphone with respect to the magnetic north. This sensor however, occasionally provides inaccurate measurements due to the presence of structural elements or electronic equipment that have a significant effect on magnetic field. Common smartphones are not equipped with other means of orientation sensing. Hence, the inaccurate magnetometer is the only way to determine the orientation of the smartphone. Since our Dead-reckoning algorithm uses the magnetometer to determine the user's orientation, such issues will lead to inaccurate readings. To tackle this problem, we take the moving average of the past ten values retrieved from the magnetometer. This enables us to minimize errors in our estimation.

Acceleration of the smartphone along the three axes, as described previously, can be obtained from the accelerometer. We use the accelerometer data to develop a step-detection algorithm and combine it with the orientation data to determine the current location of the user.

3) Step Detection Algorithm: When a user holds the smartphone parallel to his body (in a landscape orientation) while walking, we observe that the value of acceleration along the X axis varies according to a pattern. There is a steep increase in $x_{acc,t}$, as the user takes a step followed by a trough as shown in Figure 3. Therefore, by counting the number of such peaks in the $x_{acc,t}$ value, we can determine the number of steps taken by the user. However, since the sensor coordinate system is fixed relative to the orientation of the smartphone, any change in orientation of the smartphone due to user handling affects the value of acceleration along the X-axis. Even slight changes in the orientation of the smartphone result in $x_{acc,t}$ values, above our threshold for step-detection. To tackle this problem, we use the magnitude of the acceleration vector instead of monitoring the values along any particular axis. Variation of the magnitude of the acceleration vector with the threshold value is shown in Figure 4.

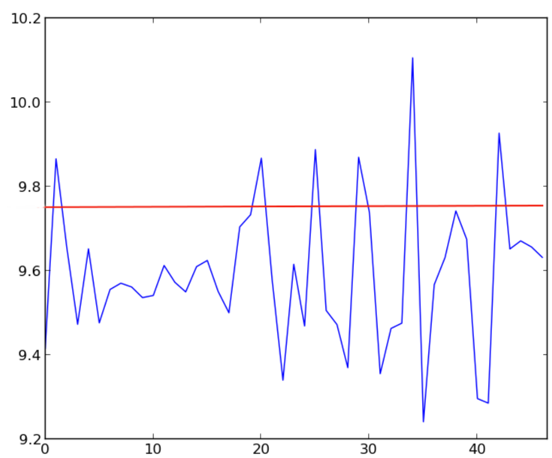

Fig. 3. Variation of $x_{acc,t}$ while walking.

We observed during experimentation that at times, two continuous peaks occur in a very small duration even when only one step is taken by the user. The magnitude of the acceleration vector for both of these peaks is higher than the threshold value and hence they are incorrectly recorded as two steps. We used the following signal processing algorithms to tackle this challenge;

a) Low - Pass Filter: A filter that passes low-frequency signals and attenuates the signals with frequencies higher than the cutoff frequency. The actual amount of attenuation for each frequency varies, based on the filter design. Hence, the filter

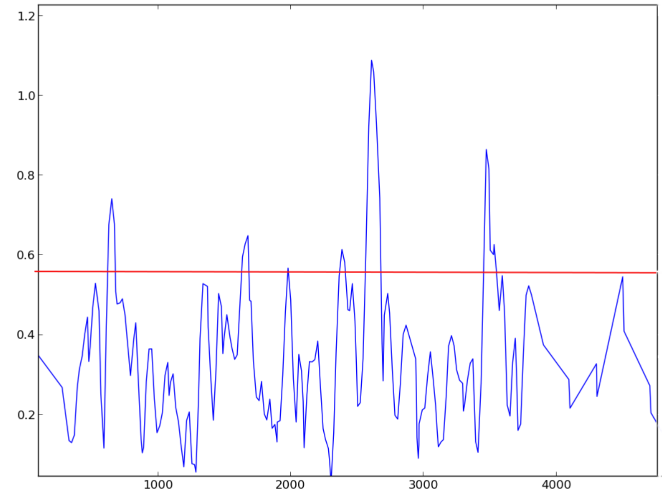

Fig. 4. Variation of $\sqrt{x^2_{acc,t} + y^2_{acc,t} + z^2_{acc,t}}$ with the threshold value for step detection

omits high frequency values, which helps in countering multiple step detection for one step taken. Algorithm 1 explains the steps.

initialization;
for $0 \leq i \leq n$ **do**
  $y[i] := y[i-1] + \alpha * (x[i] - y[i-1])$
end

Algorithm 1: Applying a Low-pass filter with α as filtering factor.

b) Step Duration Filter: While walking normally, the duration of two succeeding steps does not change too much. Hence, to increase the accuracy of our step-detection algorithm, we reject a detected peak when its duration is less than half the duration of the previous step.

4) Tying it all together; The Dead-reckoning Algorithm: On being provided the initial location, the application starts the Dead-reckoning algorithm as described in Algorithm 2. Whenever the user takes a step, the algorithm creates a set of predicted nodes. Predicted nodes are a subset of all PoIs that can be reached from the current location without any major change in the orientation of the smartphone relative to the magnetic north, i.e. nodes that can be reached by walking in a straight line. Since walking in an exact straight line would be a difficult task, all possible nodes that lie within a 30° arc from the current orientation of the smartphone are considered as predicted nodes. When the user takes a larger turn ($> 30°$) the application calculates the ratio between the weights of the predicted nodes and the current step counter value. The best ratio is then used to predict and update the node at which a turn was taken. The value of the step counter is set to zero at every such turn.

III. Implementation

The proposed approach was implemented and tested in our university campus premises. We designed an application for

initialization;
while *Step Taken* **do**
  $stepCounter = stepCounter + 1$
  Calculate current angle of device from magnetic north;
  Update list of predicted nodes;
  if *turn taken* **then**
    match ratios of edge weights and the number of steps taken to predict the best matching node;
    $stepCounter = 0$
  else
    continue;
  end
end

Algorithm 2: Localization using Dead-reckoning

the Android platform to upload the panoramic images to our Map-Maker. Using the Map-Maker we updated the tags on these images using a simple drag-to-draw mechanism. We then linked these tags to other images replicating real life indoor pathways. We also added other descriptive details to the map such as names of PoIs. These details are displayed to the user when using our navigation application.

When using the navigation application, the user can choose to view the complete indoor map of the area or select a destination location. Our navigation algorithm estimates the current position of the user on being provided with an initial location. The initial location as of now, needs to be provided by the user. However, our approach can be extended to incorporate NFC tags or QR Codes at entrances or by incorporating the last received GPS coordinate before the application switches to the indoor navigation module. On providing a starting point, the application sends a request to our web server, which replies with a JSON response. The response contains details of all the panoramas for the indoor location, along with the linking of the tags and the respective step counts.

Once the user inputs the destination, the application calculates the shortest path from the user's current location to the given destination. We use Dijkstra's algorithm to calculate the shortest path between any two nodes, taking into account the weight of the edges i.e., the step counts. This path is then displayed on the map in Blue. Figure 5 shows a screenshot from our smartphone application obtained after running the shortest path algorithm from the user's location to the given destination, A-506. The current location is also displayed to the user as a Red Dot as the user navigates through the environment.

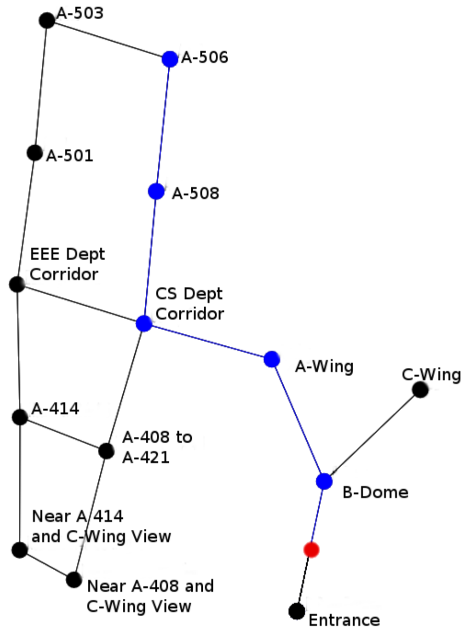

Fig. 5. Shortest path to the destination in Blue with current estimated location denoted by a Red Dot.

IV. Conclusion and Future Work

In this paper, we presented a system for indoor navigation using off-the-shelf smartphones. In the first phase, a user creates an indoor map of the area by linking panoramic images using our web application, Map Maker. This indoor map is then used by our smartphone based Navigation application to estimate a user's location, calculate the shortest path and help in navigating the user to a destination. In future work, we plan to extend our smartphone application to automate the process of step counting when capturing panoramas. We also plan to explore realtime image matching techniques using smartphones to increase the accuracy of our indoor navigation system.

Acknowledgement

This research work was conducted while the first author was an undergraduate student at BITS Pilani - KK Birla Goa Campus. The authors would like to thank the Network and Systems research group, BITS Pilani - KK Birla Goa Campus for their support.

References

[1] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: a sequence based technique for rf localization in wireless sensor networks," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, April 2005, pp. 285–292.

[2] A. Finkel, A. Harwood, H. Gaunt, and J. Antig, "Optimizing indoor location recognition through wireless fingerprinting at the ian potter museum of art," in *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, Oct 2014, pp. 210–219.

[3] Y. Li, P. Zhang, X. Niu, Y. Zhuang, H. Lan, and N. El-Sheimy, "Real-time indoor navigation using smartphone sensors," in *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, Oct 2015, pp. 1–10.

[4] S. Schmitt, S. Adler, and M. Kyas, "The effects of human body shadowing in rf-based indoor localization," in *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, Oct 2014, pp. 307–313.

[5] E. J. Hoffmann, M. Werner, and L. Schauer, "Indoor navigation using virtual anchor points," in *2016 European Navigation Conference (ENC)*, May 2016, pp. 1–8.

[6] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, "Indoor human navigation systems: A survey," *Interacting with Computers*, p. iws010, 2013.