Indoor Localization and Navigation Using Phone Sensors and a 3D Model of the Building

Szabolcs Orban
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
orbanszabolcs11@gmail.com

Teodor Stefanut
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
teodor.stefanut@cs.utcluj.ro

ABSTRACT
This article presents an innovative approach in offering indoor navigation assistance based on mobile device sensors and a 3D model of the building. We will discuss the three main stages that are required in order to implement the proposed solution: (1) creation of the 3D model, (2) definition of the navigation attributes (built to also support localization) and (3) implementation of all the required functionalities in an Android application. One of our main concerns is related to the identification of the current location of the user in an indoor environment efficiently using the device's sensors. For this purpose we will describe an algorithm that relies on the accelerometer and the compass to provide accurate location information and to simulate the movement of the user into the environment, without consuming too much energy.

Author Keywords
indoor localization, mobile device sensors, indoor navigation, virtual representation

ACM Classification Keywords
D.2.2 Design Tools and Techniques: User interfaces; J.m Computer Applications: Miscellaneous; H.5.2 User Interfaces: Interaction styles.

INTRODUCTION
In this paper we propose a solution to provide helpful user guidance through different faculty buildings. Indoor localization based on smart phones is a popular research field nowadays because of the popularity and advanced functionalities of these devices. There are three main approaches that can be used to establish user location: based on GPS (Global Positioning System), based on RSS (Received Signal Strength - the attributes of the WiFi signal received by the phone) or based on the use of data provided by other device sensors (accelerometer, gyroscope, compass, etc.).

The most famous localization technology is the Global Positioning System (GPS). Its main problem is that it requires a high amount of energy to function and drains the phone's battery rapidly. In addition, GPS signals are transmitted in a frequency that cannot easily penetrate walls and other barriers, so it can hardly be used indoor.

Another popular approach, much more energy efficient than the previous one, refers to the use of Received Signal Strength (RSS) of the WiFi signal, which can be used for location estimation based on the previously known location of the access point. This is currently considered to be a valid metric for location calculation and the vast majority of mobile devices support and implement it. The drawback of this method is that its accuracy is not good enough for guidance inside a building, and that is why it is considered to be unreliable for our purposes [10].

The newest method is related to the new internal sensors which were integrated to the current smart phones hardware. Namely, they are the *accelerometer* and the *compass*. The accelerometer measures the current acceleration of a device compared to free-fall acceleration. A compass is a sensor that indicates the north direction by measuring the three axes computed from the magnetic field strength. There are a lot of applications that use these sensors. For example, smart phone games use the accelerometer to detect the angular movement (tilting) of the device.

In his article, Moustafa Youssef [6] suggested that a good way to compute the location of the device relative to a previously known point is to use the functionalities of the accelerometer and the compass. These sensors work well using the device's resources efficiently (e.g. battery).

In previous research the accelerometer has also been used to determine human position in an environment [2]. The possible positions were classified in three main categories: walking, sitting or standing. By also using the location, the system can determine whether the person was sitting at a pub or walking in the park.

Other systems use accelerometers to detect the number of human steps. For example the pedometers are devices that count each step a person takes by detecting the motion of their hands or their hips. These devices are widely used by people in order to motivate themselves into exercising routines or just to increase their physical activity [9]. Pedometer based applications count human steps using different movement patterns and thus are able to determine the distance traveled. The information registered is later matched with the direction obtained from compass data.

In a similar way, our proposed approach aims to combine the data from the accelerometer and from the compass in order to define the movement of the mobile device in the indoor environment and to reflect these changes into our 3D building model. While we use the compass to establish the direction of motion the accelerometer allows us to estimate the distance traveled by the user on the detected direction.

RELATED WORK

Indoor Localization

Although GPS-based approaches can provide a good performance in outdoor scenarios, in the case of indoor localization the system will have some issues regarding the accuracy of the result.

In [3] Kaiqing Zhang described a positioning method based on WiFi RSS that enables the estimation of indoor location using smart phones. In order for this approach to work as expected, map information about the environment is also necessary for measurements and real-time indoor localization.

WiFi RSSI (Received Signal Strength Indicator) approach indicates the distance between the user and the access point to which the device is connected. The advantage of this approach is that WiFi technology is found almost everywhere and no additional infrastructure is needed for localization. There are two main approaches based on this technology: fingerprinting [4] and ranging-based methods [5]. The localization algorithms use combined location information from IMUs, maps and WiFi RSS. In order to improve the performance, the map is represented using area states and for each state a different localization approach is implemented.

3D Navigation

This is considered to be a new way of using a normal map, as 3D technology offers not only a representation of the real-world, but also gives users relevant information as advanced city models, 3D landmarks and icons or a digital elevation model [8].

In order to create a 3D Navigation model, there are three main aspects that need to be taken into consideration: to represent the navigation environment, to support path finding and to provide context information about the location. Each of these requirements can be fulfilled by using a special 3D model which contains both topology and semantics of the building. Next to the model created, the system also requires a representation of the topology of the building in the form of a network and information about how is this related to the geometrical model.

In order to handle the movement of a user inside a 3D model there are some problems that need to be taken into consideration. One solution proposed by W. Xua and M. Kruminaite [11] is to define a pattern so that the accelerometer measurements can be interpreted as movements. The solution also uses step detection methods to monitor how the acceleration value drops when the user is balancing the phone while walking. Also the turning can be determined by checking the abrupt change of the azimuth angle given by the compass. By using these types of sensors the location data will not be affected by the electrical appliances and magnetic materials from indoor locations.

There are no standard representation rules for defining all elements from an indoor environment, so the system needs to have its own rules. In other words the map should be represented using intuitive and easy to understand representation rules. The model should contain semantics that represent the navigation environment and the context information.

A good way to achieve this type of model is the method used by OpenStreetMap (OSM) [1] which is an open source geographical database. The structure of the network used consists of nodes and relationships (edges). Nodes are spatial components and edges represent the connections between different components (see Figure 1). Each relation can get an annotation and a label by using a key-value-pairs structure. The tags are used to define the elements from the map that are represented in the 3D model and provide a better understanding on how navigation is performed.

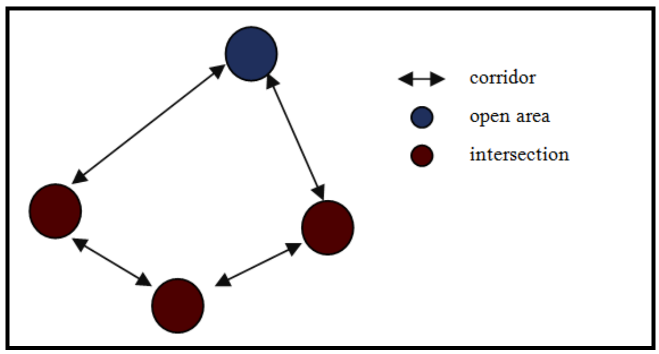

Figure 1: The graph representation of indoor elements.

Each node can contain different elements in order to be classified. For example an intersection point can be defined as a "door" or a "turning point". In indoor representations, these elements show an entrance to an open area or the link between two corridors. A corridor can have doors and walls defined within its elements [3].

There are four different actions taken into consideration depending on the above defined model:

1. *Walking in Corridor*: because the width of one corridor can be relatively small, the user moves in one dimension while walking in the corridor.
2. *Turning*: the user is at a turning point. When he is close to a turning point on the map, an abrupt change of the compass angle is detected.
3. *Verifying the turning*: it is possible that the chosen corridor may not be the right track.

4. *Walking in Open Area*: this action is determined by the fact that the user walks into a "Corridor door" that connects a door with a room.

The network is bound with the 3D geometrical model using different IDs for nodes stored in the object's attributes.

PROPOSED METHOD

The proposed application has the role of a guide in the faculty buildings. It allows any user that possesses an Android phone to search for different classrooms, administrative locations or laboratories.

As an initial step, the user must specify to the system his current location and the name of the classroom he wants to reach. The application displays the 3D model of the building marking the user's current position and an optimal path to the destination.

In order to provide accurate information and synchronize the virtual movement to the real one, the application has to detect the progress of the user and navigate him to the classroom. He/she has the possibility to move freely in the building while the application maps his position from the physical world to the virtual world model in real time.

3D Modeling and Loading the Model

Creating a 3D replica of a building is not a very simple process. First of all the exact blueprints of the structure are needed and must be loaded into a 3D modeling program. After the prints are loaded all the lines of the walls must be traced keeping gaps for the doors.

After the tracing of the walls on a 2D plane, the lines must be extruded to a specific height to form a 3D wall. The next step is to fill the upper side of the doors until the roof is reached to form a door-like gap that is smaller than the height of the wall.

In the case when we have to model more than one floor, except for the base of the upper and lower levels we also have to model the stairs connecting these or even an additional semi-level between the two floors. This part also needs to be modeled with the exact data from the blueprints.

Finally texture must be mapped onto the model and exported into a format which can be read by Android Studio.

Graph Path

One of the core functionalities of the application is to compute the shortest path from the current location of the user to the indicated destination. In contrast with the graph building strategy presented in [3], our approach does not classify the nodes of the graph depending on the type of the room. In the application there is no need to classify them, because there is no variety of rooms.

In order to compute the shortest path a predefined graph containing the coordinates of the rooms and some connector nodes are used. After the optimal path is computed the result is shown on the loaded 3D model as shown in Figure 2. Furthermore the user is free to walk wherever he/she pleases, as the current location on the graph is always computed to the closest node.

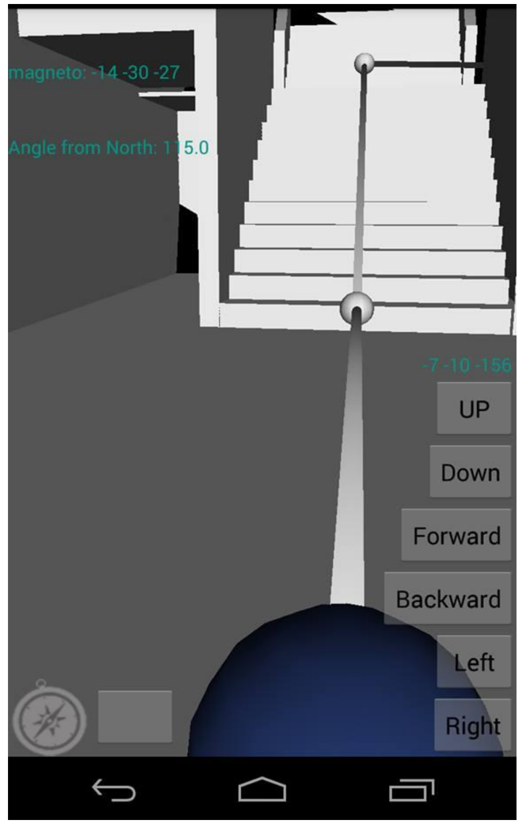

Figure 2: Example of the 3D model and the graph path

From time to time, the system may accumulate some positioning errors. These errors appear when the movements performed by the user in the real world are not correctly detected by the device. For correcting these errors the user can adjust his present location by jumping to the previous or to the next node and continuing the navigation from that position.

Identifying the correct location in the virtual model can be achieved by the user based on the similarity of the model to the real world. By "looking around" in both environments, the user should be able to identify the closest marker to his real position or to realize he/she leaved the path indicated by the system. A very important role in the accuracy of user decision is played by the photorealism of the 3D model.

Rotation Interaction

The main interaction types with a 3D object involve observations and rotations in all the available directions. On a smart phone this actions are usually achieved by swiping to a direction.

The physical screen has only two coordinates: X and Y (see Figure 3) while the system of the model has three coordinates: X, Y and Z (see Figure 4). The main challenge is to map the actions from the 2D screen to the 3D space.

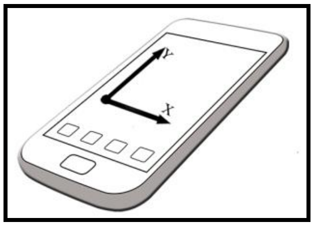

Figure 3: Coordinate system of the mobile screen

Figure 4 shows that the model can be rotated in three directions, but because of the representation constraints rotating the model around the Z axis is not recommended (as it would allow the user to turn the buildings upside down). Therefore for the rotation only the two virtual axes, X and Y are used. On the other side, the coordinate system on the screen of the phone is a Cartesian coordinate system: X is the horizontal and Y is the vertical coordinate (see Figure 3).

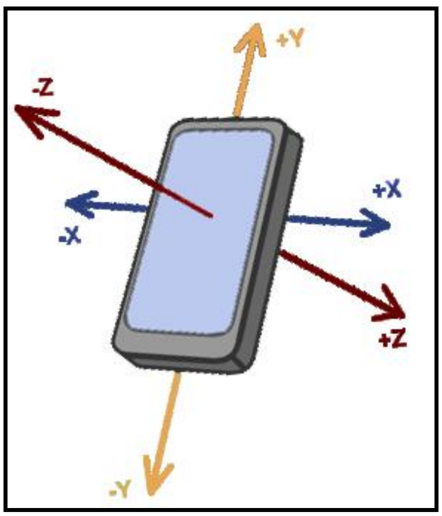

Figure 4: Coordinate system of the virtual model

Touch detection is checked and saved periodically. The mapping of the physical coordinates to the virtual ones has the formula:

$$R_X = R_X + (touched_Y - current_Y) \times K$$

$$R_Y = R_Y + (touched_X - current_X) \times K.$$

R_X and R_Y are the Euler rotation angles of the virtual X and Y axis of the model. The two float values $touched_X$ and $touched_Y$ are the coordinates of the point where the touch began, respectively the values $current_X$ and $current_Y$ compose the current position of the user's touch on the screen. The value K is a constant which is responsible for the correct use of scale.

Gyroscope Rotation

Another possible way of rotating the object is to use the built in gyroscope of the device. The gyroscope is a sensor that detects the rotation around the three axes. The value returned is linear to the change rate of the angles and is expressed in deg/s.

The degree can simply be computed from

$$degree = \frac{G_1 - G_0}{t_1 - t_0}.$$

The gyroscope value is G_1 at time t_1 and G_0 at time t_0. Finally the result can be added to the rotation properties of the object.

This feature can be used without swiping or touching the screen and allows the user to rotate and to look around. To correct the possible detection failures of the gyroscope, the user can easily adjust the angle with the swiping interaction.

Accelerometer

For computing the distance traveled by the user in the real environment, the use of the built in accelerometer is needed. This sensor detects the force directed to the opposite direction of the acceleration vectors.

To get the sum of the three vectors

$$V_X, V_Y, V_Z,$$

the following formula must be used:

$$V = \sqrt{{V_X}^2 + {V_Y}^2 + {V_Z}^2}.$$

If the device is situated in a motionless state the force of gravity still applies to the Z axis of the accelerometer. This force must be constantly decreased from that vector.

In Android there are two types of accelerometers already implemented. One is the TYPE_ACCELEROMETER, which is a hardware type sensor used for motion detection as tilt or shake of the device [7]. This is the regular sensor already presented. The other one is called TYPE_LINEAR_ACCELERATION, which is a hardware and software type sensor. This sensor returns the acceleration force applied to all the three axes excluding the force of gravity. It is used for monitoring the acceleration.

The second sensor is suitable for the application and therefore this one is chosen. In the analog world continuous math is used to determine the distance travelled which is:

$$Velocity = \int Acceleration$$

$$Distance = \int Velocity$$

In the digital world discrete math can be used and the integration becomes summation:

$$Velocity = \sum Acceleration$$

$$Distance = \sum Velocity$$

Step Detector

For illustrating the movement of the user in the virtual representation of the building, the implementation of a step detector is required. As mentioned before, this can be achieved using the accelerometer.

Each time the acceleration is triggered the detector decides from the force of the movement whether it was a step made by the user or just a hand movement. If a step is detected the application performs the translation of the 3D model according to the movement direction.

To eliminate the possible noises, a low-pass filter is applied to the detector. This hinders data from small movements of the hand. At the same time, a high-pass filter is also implemented to prevent shacking and sudden movements from being considered as steps.

Compass

In order to bind the movement of the user with the movement in the 3D model an orientation method is needed. A simple and efficient way to achieve this is to use the position of the North Pole associated with a fixed point in the 3D world. The orientation of the virtual user can be computed from the angle between the phone's orientation vector and the point of the North Pole.

Using the device's geomagnetic field sensor the changes of the earth's magnetic field can be monitored. This sensor returns the azimuth, pinch and roll values. For the compass to work only the azimuth value is needed. Azimuth indicates the rotation around the axis that points towards the center of the Earth.

Knowing the angle towards the North Pole and the distance travelled from the accelerometer's data, the device is able to synchronize the position of the user in real world with the position in the virtual 3D world, independent of the movement trajectory or the virtual looking-at direction.

Phone Holding Anomalies

The previously shown solutions work only if the user is holding his device straight vertical to the ground because of the orientation of the used vectors. In real usage scenarios however, the average user holds the phone oblique around the X axis as shown in Figure 4. Additional tilting of the phone during the use causes the device to register incorrect values along the Y vector, affecting the computed results.

For most of the scenarios, because the Y vector is not part of the main component of the straight movement, it can be usually ignored.

Stair Problem and Solution

Ignoring the Y vector does not lead to problems when walking on a single floor. However, it causes problems in case of going up the stairs. As the device cannot detect the vertical movement of the user, it cannot correctly represent the height position in the virtual world.

However, if the optimal path contains a staircase, then the application can access the 3D positions of the nodes before and after the stairs. A quick and easy solution for positioning the user correctly into the 3D model is to compute the height of the corresponding position on the stairs based on the two known coordinates and assign it to the height of the user's position.

The algorithm starts when the user has reached in his/her path a segment defined by two nodes that have different height values:

1. Get the closest point on the segment from the user's current position using only the X and Z values
2. Compute the height of the point as a linear interpolation between the segment's end points
3. Assign the height of the point to the height of the users position

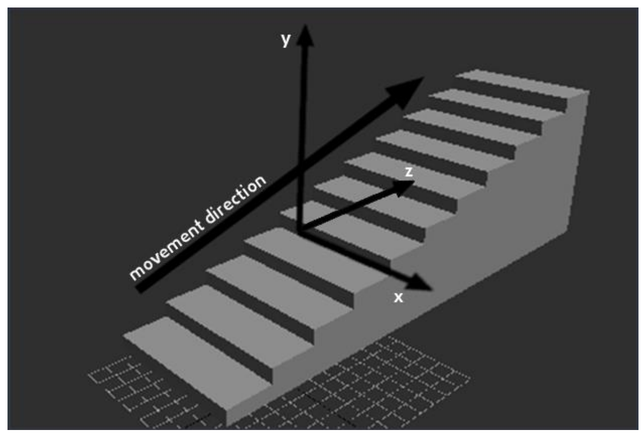

Figure 5: Computing the height position on the stairs

This way the positioning of the user in the virtual world always corresponds with the real position on the stairs.

USER SCENARIOS

At the beginning, the user must indicate his current location from a predefined list and the room number he seeks, as shown in Figure 6. This step is necessary due to the implemented localization approach, which (as mentioned in the previous sections) requires a previously known location for being able to correctly compute and map the user movement.

After retrieving the information from the user, the application will search for the optimal path between the two locations, based on the physical distance. Finally the user will be presented with the 3D model of the building and the highlighted route to be followed (see Figure 2). The initial position of the user into the virtual world is established at the predefined coordinates associated with the initial location selected.

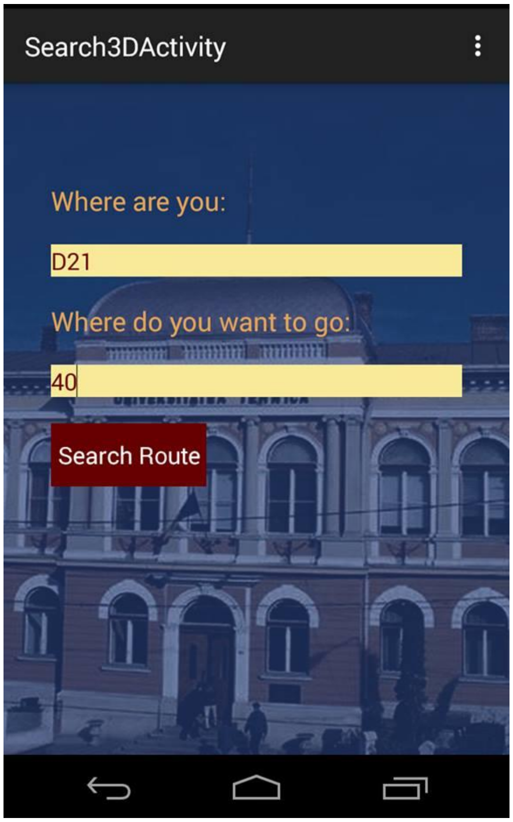

Figure 6: The initial step in finding the desired route

After the map is displayed, the user can interact with the virtual world through swipes and dedicated UI controls. The application functionalities are represented in an interface created using the graphic layout tool for Android so that anyone can easily use it. In Figure 7 the main flow of the application is represented graphically.

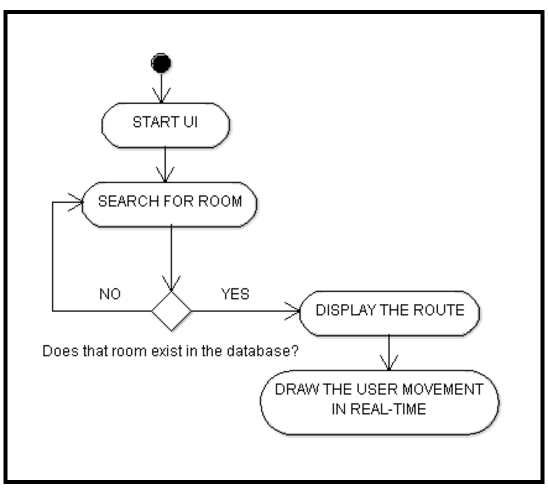

Figure 7: Main flow of events when searching a room

CONCLUSION

The result of our work is a stand-alone application that is used for real-time indoor localization and navigation based on a 3D model improved with map-like information.

Also, in order to get the location accurately and without consuming that much energy of our mobile device we implemented a method which uses only the phone sensors (accelerometer and compass) to determine the location and movement of the user. This is considered to be a new approach in handling indoor positioning.

Specific user interactions have been implemented to enable easy corrections of the computational errors related to the synchronization of the user location in the virtual and in the real world. The method by which the user interacts with the application is a good example of a prototype and can be used in future similar applications.

REFERENCES

1. *Indoor Mapping – OpenStreetMap Wiki*, Sept. 2014. [Online]. Available:

   http://wiki.openstreetmap.org/wiki/Indoor_Mapping
2. Jim Scarlett, *Enhancing the Performance of Pedometers Using a Single Accelerometer*
3. Kaiqing Zhang. Hong Hu. *Indoor Localization Algorithm For Smart phones*
4. P. Bahl and V. N. Padmanabhan, *RADAR: An in-building RF-based user location and tracking system*, in *IEEE Annu. Joint Conf. on Comput. and Commun. Soc.*, vol. 4, 2000, pp. 775–84
5. P. Bahl and V. N. Padmanabhan, *A comparative study on indoor localization based on RSSI measurement in wireless sensor network*, in *IEEE* Int. Joint Conf. on Comput. Sci. and Software Eng., Nakhon, Pathom, May 2011, pp. 1–6
6. Moustafa Youssef, Mohamed Amir Yosef, *GAC: Energy-Efficient Hybrid GPS-Accelerometer-Compass GSM Localization*
7. Sensors Overview, Android Developers

   http://developer.android.com/guide/topics/sensors/sensors_overview.html
8. Tomtom 3D Navigation

   http://automotive.tomtom.com/en/real-time-maps/3d-navigation
9. Tudor-Locke, *Taking Steps toward Increased Physical Activity: Using Pedometers To Measure and Motivate*
10. Ubejd Shala , Angel Rodriguez , *Indoor Positioning using Sensor-fusion in Android Devices*
11. W. Xua, M. Kruminaitea, B. Onrusta, *A 3d Model Based Indoor Navigation System For Hubei Provincial Museum*