Indoor Localization Using Bluetooth

Tiago Fernandes,

Faculdade de Engenharia da Universidade do Porto,
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
pro10018@fe.up.pt*

Abstract. This article proposes a solution for user localization in indoor areas using the compass, accelerometer and Bluetooth to calculate the user's position within this environment. This application is viewed through a 3D virtual environment representing a simple room. The basis of this solution is the utilization of a mobile Bluetooth-enabled device, such as a PDA, where the application is deployed. The application will then use the mobile phone's Bluetooth to determine the Received Signal Strength Indicator (RSSI) of beacons located within the area. This information is then used to determine the virtual position of the user by triangulation. Additional sensors, such as the accelerometer or the compass provide extra precision and compensate the latency that the Bluetooth positioning solution provides. This solution has proved to be accurate, inexpensive, and very usable, as it uses virtually no input from the user (since the input the user provides is actually passive). Also, it is tolerant to errors and does not conflict with any other Bluetooth devices, such as other mobile phones.

Keywords: Mobile, Indoor, Localization, Bluetooth

1 Introduction

Having accurate information about people's location in indoor environments is crucial for some applications such as e-commerce and e-museums. Solutions such as GPS and GSM location systems, for example, are very inefficient when used inside buildings. Acquiring relaying hardware that could enhance the signal inside these areas, therefore granting a better performance for the location, could solve this problem. However, the required components are usually very expensive, especially since the necessary number of relays would vary according to the area being covered.

This article proposes a low-cost solution for creating a system that is able of providing indoor-location information using Bluetooth and modern handheld devices. The idea behind this project is based on the utilization of small, inexpensive Bluetooth devices that are placed on the area to be covered. These devices are registered on the handheld during a calibration phase and are used for triangulation. However, to enhance system accuracy, the mobile device used for location may also contribute by providing information from other sensors such as accelerometers and digital compass. These extra bits of information are weighted and mixed together to produce a final calculation over the user position and orientation. Also, this system is targeted to achieve several goals besides the indoor localization of users, such as maintaining user privacy and provide near real-time information, since all the calculations are decentralized from the infrastructure and performed directly on the mobile device. This guarantees that no personal or private data is stored in external servers, since the external Bluetooth devices are only used to broadcast their own address.

This article starts by presenting an overview of the existing technologies in section 1.1. The methodology used for this work is presented in section 2, which is divided in three parts, namely for calibration, localization and alternative data sources in sections 2.1, 2.2 and 2.3, respectively. The obtained results are presented in section 3. Finally, section 4 presents the work conclusions and perspectives of future work.

1.1 Previous work

Some work has been developed for indoor-location, using several distinct technologies. Veljo Otsasson et al. [1] conceived a system that was able of providing user position in inside environments using GSM triangulation. The idea behind this project is to use wide fingerprinting that uses GSM cells that are strong enough to be detected but too weak to be used in communication, in addition to the six cells defined in the GSM standards. This system has many advantages such as the range of signal coverage, the fact that any mobile phone could be used for positioning and that the system would be highly tolerant to power shortages. In order to be able of detecting the user position accurately, this system requires some calibration that was performed by measuring both the 802.11 and the GSM signals in each division of the tested areas. By using the proposed algorithms that held the best results, this solution was able of reporting the user location with a median localization error between 2.5 and 5.4 meters.

In a different perspective, the Cricket project [2] uses both Radio-Frequency (RF) and ultrasound signals to identify a user's position. The utilization of both sensors is based on the fact that RF propagates in non-linear and possibly unpredictable ways inside buildings. Therefore, it was necessary to consider alternative ways of providing increased precision to the position calculation. So, to perform the calculations, the beacons send concurrently RF and ultrasonic signals. As the speed of sound is smaller than the transmission speed of RF signals, the later will arrive sooner to the listeners. When a listener receives a RF, it uses the first bits as training information, enables the ultrasonic receiver and waits for the ultrasound emitted by the beacon. The calculation is then performed by using both the strength of the RF signal and the time difference between the arrivals of each signal. One of the great advantages of this project is the low cost that is required to buy all of the components. The error rate reported for mobile devices is however, somehow big, being around 20-25%.

HP also developed a solution that uses infrared beacons instead of ultrasound and typical RF emitters, called HP Cooltown [3]. To find its location, the user must point its infrared-enabled handheld device to the infrared beacons. This has the clear problem of requiring user interaction to work, but on the other hand, this method also protects the user's privacy, since he only interacts with the system when he really wants to.

Finally, F. J. González-Castaño and J. Garcia-Reinoso [4] developed a system that attempts to provide user location in indoor environments using only Bluetooth devices. This proposal uses a network of bluetooth devices, organized in hexagonal grids. Each node is either a slave or a master node. The user is equipped with a Bluetooth enabled or Bluetooth badges and broadcasts its address to the nodes. Every slave node receives the RSSI value from the user and sends it to the master node. The master node performs every calculation to triangulate the user position based on the RSSI values that were received as well as the slave nodes positions and sends the computed data to some servers that will use that information for some service. This approach is very expansible since the system is able of auto-configuring itself automatically. Also, there are no collisions with other existing devices, because the work is centralized on the slave and master nodes which conform to a specific protocol. However, given that all the calculations are done by the master nodes the system may become quickly overloaded which brings performance problems in terms of response times, depending on the number of position calculations and Bluetooth devices in the network.

2 Methodology

The first step towards the resolution of this problem was to create a solution that was able of receiving any type of sensor data and return a position. For that, it was necessary to have some calibration results from the sensors, so that the range of values was known and the distances that those values correspond to. By doing so, it is possible to compute a linear calculation based only upon these two points. However, if possible, some intermediate values could be used for a more precise interpolation, if needed, increasing the overall accuracy of the solution. Therefore, prior to developing the solution itself, a calibration tool was needed.

2.1 Calibration

A simple calibration tool was developed with the single purpose of selecting the sensors that will be considered by the application, so that it does not conflict with other Bluetooth devices. This simple application finds every Bluetooth device that is detectable in the area that is being tested and lists it on the device. To perform calibration, the user must select each of the relevant devices and save the RSSI value for each distance that is to be used. The values saved in the calibration tool will then be exported in an XML file that specifies which sensors are to be used and also some other information that is to be used within the application, such as the position of points of interest (POI) and the 3D model filenames.

It is important to notice that the calibration distances are not fixed. The granularity of the measurements and the distance values that are to be used in calibration may vary according to the environment and the Bluetooth devices being used. This calibration step only needs to be performed once and the application is ready to be used with the same XML file in every other run, unless maintenance is required.

2.2 Localization

To find the user's location, the application starts by an initialization phase in which the data that was saved in the XML file is read. As mentioned above, the XML file contains a set of RSSI-distance value pairs for each sensor. These values are read onto a hash map during the initialization. When the application loads, the Bluetooth receiver is enabled and the device starts to look for nearby known devices that have addresses that were registered within the XML file during the calibration. If at least three values are found, the software uses a triangulation algorithm that uses the obtained values and calculates the user's position. For each of the sensor values found, the application searches the hash map for the calibration points read from the XML file and finds all the intervals in which the obtained sensor value is contained. Since the RSSI variation is much greater when the device is near the Bluetooth beacon [5][6], the algorithm weights the possible found intervals by giving slightly more relevance (~10% more) to the calibration values that are more distant.

Each of the RSSI values obtained represents the radius from a circle, whose center is the position of the sensor that originated that value. In a first step to perform triangulation, two of those RSSI values are used, as well as the positions of the respective sensors. If the circles intersect then three points are saved, namely the two intersection points and the point where the line defined by the circle centers crosses the line defined by the two intersection points. With these three points saved, the algorithm searches for a third sensor that defines a circle that intersects the previous two. To do so, it simply checks if any of the intersection points calculated previously is within range from the sensor's radius. If that is the case, the intersections between the first and third circle and between the second and third circle are calculated. Finally, the midpoint of each of these intersections is used to calculate an average for the user's position.

There are two clear problems with this algorithm. The first problem appears if one of the circles is too small, there will be no intersection with any other two circles. If this happens, it means that the user is actually very close to the sensor itself and that is the reason for such a small radius. To solve this case, the user's position is snapped to the sensor position as long as it is within the map-defined bounds. The precision of the value that triggers this snapping behavior is controlled by a parameter that can be set in the application configuration. However, there is a second problem. Even if the circle is not sufficiently small to cause the user's position to be snapped, there might be times where one circle is contained inside another circle. These cases happen when the user is relatively close to one sensor, and far away from other sensor that has a very wide range. Also, this might also happen when the sensors are too close to each other and the user is close to one of the sensors. This causes a logical problem since the intersection points can't be calculated. But, since this means that the user is close to one of the sensors, the solution is to ignore the intersection calculation (since it wouldn't be possible, anyway) and consider that the central intersection point is the center of that circle.

The final position is calculated by using the values obtained from the hash map inside the triangulation algorithm. This algorithm produces a final set of coordinates based on the interpolation of the three points that were found, if any, using the method described above. This process guarantees that the coordinates conform to the specific environment, in a somehow pessimistic form, since the farthest values have more weight than the closest.

2.3 Alternative Data Sources

Since the Bluetooth takes some time to obtain the updated RSSI values, it was necessary to compensate the utilization of these values in the meantime with values from other sources. The modern PDAs and mobile phones are usually equipped with accelerometers and digital compass. By using the accelerometer data, it was possible to develop a rather sensible pedometer that indicates if the user has walked. Additionally, the accelerometer data also allowed for the creation of a module, capable of determining the 3D orientation of the device. Mapping this orientation of the device with the OpenGL camera, allowed for a pseudo augmented reality interaction with the virtual environment. Note that the user is only providing passive input to the device, meaning that virtually no experience is needed with the handling of the device, or the application. The walking direction is given by the digital compass and introduced into the application. So, in case the Bluetooth fails or the handheld takes a long time to receive RSSI values, the user position is updated with the information provided by both the pedometer and the digital compass.

This raises a significant problem that can't be overlooked. With the pedometer it is possible to know if the user walked and the compass provides the direction he was facing. However, it is impossible to know if the user walked backwards. Theoretically, this could be read from the accelerometer values, but due to the variations that are usually read, it would be very difficult to know accurately if the user actually moved backwards. In these cases, the only solution is to wait for the Bluetooth triangulation to reposition the user to the correct position, or try to use other sensors such as Wi-Fi or even the camera, although this is out of the scope of this project.

3 Results

To test this project, an application was built to reproduce the user movements inside a 3D environment, by using an HTC HD2, running a custom Android build and OpenGL ES [8]. The virtual camera placed on the scene turns automatically its direction to face the correct direction when the user moves the mobile phone, by using values read from the accelerometer and compass values. These values are submitted to a simple filter to avoid unwanted noise and to increase precision. The device was calibrated for each of the sensors, registering its Bluetooth address and the values at several distances that were previously marked. These results are presented on Figure 1, for one of the sensors, in terms of dBm and distance. The distance is measured in centimeters and, for convenience the Bluetooth dBm values are represented with its absolute values.

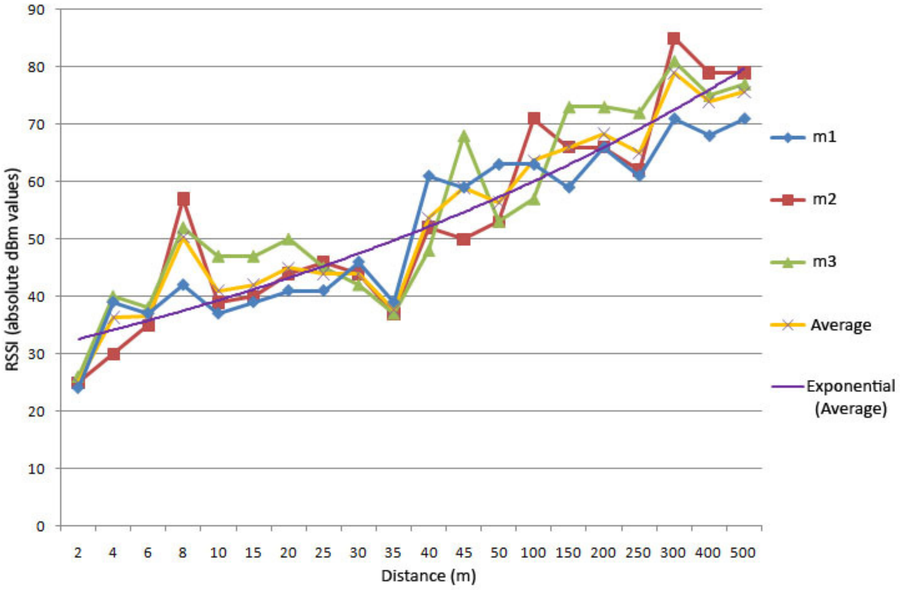

Fig. 1. RSSI values read during the calibration phase, according to several different distances in three distinct measurements. The graphic also presents an average calculation for the values and an exponential curve calculated from the average points.

The tests took place in a room with 71 m^2 with a near-square shape. This room was modeled in 3D and exported to the .obj format in order to be easily imported by the application. In the virtual environment, the whole scene transforms correctly according to the angle in which the user is pointing the device with minor differences of ~2 degrees. Also, the pedometer works very well if the model has the correct dimensions or the step size and the sensitivity are correctly configured for an average person. Even without using Bluetooth localization, if the starting points for both the virtual scenario and the user are aligned, the system is able of tracking the user with great precision (< 1 meter, for a correct step and sensitivity calibration). The room scheme and the approximate Bluetooth beacons positioning is presented in Fig. 2.

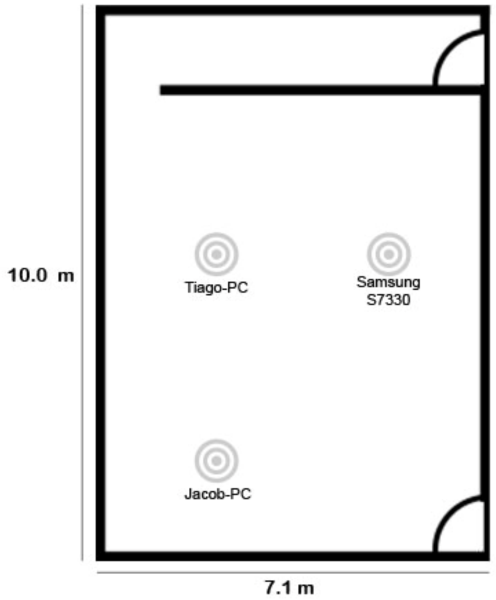

Fig. 2. Room scheme and Bluetooth beacon approximate positioning.

The Bluetooth location system is able of positioning the user correctly in a scene. However, during the testing sessions, some noise was registered making the virtual camera jump from one location to another at some times. This effect was reduced by using noise filters during signal capturing and also by using the position history to infer the probable position of the user. This avoids all of the signal peaks that are sometimes registered due to noise and signal reflection, but are still inefficient when the signal varies more noticeably for longer periods. Such problems happened mostly when using a Samsung S7330 as a Bluetooth beacon, since the signal emitted by this device is very unstable and has very large fluctuations. However, although the test space was relatively small (and the number of Bluetooth devices was also small, since there were only three testing units) the positioning with general Bluetooth devices does not have any type of collision problem. This happens because the addresses are registered during a calibration phase, and provide indoor-localization with high accuracy (~1.5 meters) when combined with another sensors. The final results are shown on Table 1. The accuracy for each method in each case is presented on Table 2.

Table 1. Obtained distance measurements using different methods and devices

Reading	Linear (m)	Inv. Quad. (m)	Realistic(m)	Device	Real Dist. (m)
1	0.28	0.83	0.306	Jacob-PC	0.40
1	3.73	4.34	3.5	Tiago-PC	3.20
1	4.68	4.41	3.6	Samsung	4.0
2	0.32	1.26	0.3167	Jacob-PC	0.40
2	3.73	4.34	3.5	Tiago-PC	3.20
2	4.89	3.89	3.0	Samsung	4.0
3	0.32	1.26	0.3167	Jacob-PC	0.40
3	5.41	5.57	5.0	Tiago-PC	3.20
3	5.32	6.33	4.25	Samsung	4.0
4	0.32	1.26	0.3167	Jacob-PC	0.40
4	3.73	4.34	3.5	Tiago-PC	3.20
4	4.89	3.89	3.0	Samsung	4.0

Table 2. Accuracy percentage for each of the different methods and devices

Reading	Linear	Inv. Quad. (m)	Realistic(m)	Device	Real Dist. (m)
1	70.0%	48.2%	76.5%	Jacob-PC	0.40
1	85.7%	73.7%	91.4%	Tiago-PC	3.20
1	85.5%	90.7%	90.0%	Samsung	4.0
2	80.0%	31.7%	79.2%	Jacob-PC	0.40
2	85.7%	73.7%	91.4%	Tiago-PC	3.20
2	81.8%	97.3%	75.0%	Samsung	4.0
3	80.0%	31.7%	79.2%	Jacob-PC	0.40
3	59.1%	57.5%	64.0%	Tiago-PC	3.20
3	75.2%	63.2%	94.1%	Samsung	4.0
4	80.0%	31.7%	79.2%	Jacob-PC	0.40
4	85.7%	73.7%	91.4%	Tiago-PC	3.20
4	81.8%	97.3%	75.0%	Samsung	4.0
Average	79.2%	64.2%	82.2%	---	---

The above tables represent a usual situation where the mobile device is able to pinpoint its real location thanks to some Bluetooth devices nearby, recognizable thanks to the XML file. The four readings were made without moving either the mobile phone or the devices. However there was a considerable discrepancy in the third reading. This was due to a person being between the mobile device and two of the beacons, effectively altering the read values. Still, these values were actually very accurate. The latency between readings would vary between 5 and 10 seconds, varying from 2 to 1 Hz refresh rate respectively, which is acceptable, for indoor navigation. It is important to notice that the measurements took place in a room with 5 to 10 persons, moving around without any pattern. During the tests, some other Bluetooth devices entered the room or walked nearby. These facts created some noise in the positioning and beacon detection but not sufficient to invalidate the calculation. This was mostly due to the fact that a medium number of points (20 RSSI-distance pairs) were used in the calibration.

4 Conclusions and Future Work

The proposed methodology described above achieves the goal of providing a simple, inexpensive and ubiquitous indoor localization solution. It is capable of pinpointing the user's location with reasonably high accuracy. Also it provides an alternative form to bypass the Bluetooth location technique's high latency by using the accelerometer as a pedometer.

The solution described above heavily depends on a calibration phase that could require too much time and effort to perform without a tool designed for that purpose. However, the utilization of the small calibration software greatly reduces the required time and expertise to configure the system, making it accessible to users without great computer knowledge and proficiency. The application is not designed to be auto-configurable, since it needs to know at least which Bluetooth sensors shall be used and two RSSI values for two given distance points to perform the triangulation. After the calibration has been done, there was no need to perform it again, even when other Bluetooth devices entered the area. Since the application knows which addresses shall be considered, there are no possible collisions between other Bluetooth devices. There exists, however, some noise due to the influence of other Bluetooth emitters, creating some fluctuations in the readings. These fluctuations are also noticeable when there are more persons in the room, increasing when they are moving, since this affects the reflection of the signals. Nevertheless, this problem proved to be irrelevant due to the fact that in most situations, the PDA is in range of more than three well-known Bluetooth devices, therefore using all of the values to compensate for eventual errors. Even when the application is run with only three sensors, the effect of the noise induced by people and from other Bluetooth devices greatly depends on the calibration. Finding more RSSI-distance pairs during the calibration makes the application more reliable and error-resistant.

According to [7], the RSSI values are more inconstant and vary greatly and in possibly unpredictable ways depending on the environment and devices being used. Instead, the Bit Error Rate (BER) or Link Quality (LQ) metrics should be used for greater precision. Yet, this was not possible, since the underlying software did not provide any access to these indicators and therefore, RSSI had to be used instead. Attempts were also made to use the iOS Bluetooth features to develop to an iPhone, but it also wasn't possible to obtain access to the BER and LQ information. From the methods depicted in Tables 1 and 2, the linear method also has a reasonable performance, but is much more susceptible to the signal fluctuations induced by noise and possible reflections. The inverse of quadratic function presents the worst results, although in some cases the values are much better than the other methods and this method performs better when the user is farther from the beacons. The realistic method, which calculates the position based on several points obtained during the calibration held the best results, with an average accuracy of 82.2%.

Furthermore, the user's interaction with the application is greatly simplified, as the user needs only to aim the phone at the areas from which he wishes to receive more information from. This has been proven to be quite a successful feature, although it was only tested with few individuals. Some of these individuals didn't possess any kind of technological background. However, all of them were able to grasp the applications concept and the ways to use it with no problem. Further testing of the application will be required in order to fully validate the interaction paradigm. Additionally, an increase in performance and precision is also considered needed, in order to make the user's experience as seamless as possible.

This solution could be improved by using computer-learning algorithms to increase user localization precision, especially when the Bluetooth signals are weak and/or combined with accelerometer information. The software could use this method to learn how much the virtual position should be changed according to the accelerometer values, when the application is waiting for new Bluetooth RSSI values. Also, this information could be complemented with a better filter for the accelerometer and prediction algorithms, typically used in network games, to compensate for possible loss of signal.

References

1. Otsason, V., Varshavsky, A., Lamarca, A., Lara, E.D.: Accurate GSM Indoor Localization. UbiComp 2005, 141-158 (2005)
2. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. ACM MobiCom (2000)
3. Kindenberg, T., Barton, J.: A Web-Based Nomadic Computing System. Hewlett-Packard (2000)
4. González-Castaño, F.J., Garcia-Reinoso, J.: Bluetooth Location Networks. IEEE GlobeCom'02 (2002)
5. Hightower, J., Borriello, G.: Location Systems for Ubiquitous Computing. IEEE, 57-66 (2001)
6. Dishongh, T.J., McGrath, M.: Wireless Sensor Networks for Healthcare Applications. Artech House, Massachusetts (2010)
7. Ørbæk, P.: Positioning and Location Technologies, http://www.daimi.au.dk/DIS/materialer/positioning-tech.pdf (2005). (last viewed on 12/12/2010)
8. Android: The Developer's Guide, http://developer.android.com/guide/index.html (last viewed on 21/12/2010)