International Journal of Smart Home
Vol.7, No.5 (2013), pp.9-18
http://dx.doi.org/10.14257/ijsh.2013.7.5.02

A Smartphone Indoor Positioning Method

Jaegeol Yim

Dept. of Computer Engineering, Dongguk University, Gyeongju, Korea
yim@dongguk.ac.kr

Abstract

Location based services are very useful to human beings. The vehicle navigation service that is used by almost all drivers in the world is an example location based service. As the man made constructions are getting bigger and bigger, indoor location based services are demanded. Matter of fact, researches for indoor location based service started in the early 1990s and developing techniques with which universally available efficient indoor location based service systems can be built is one of the hottest current research issues. Considering the features of smart phones, its computing power and the sensor devices equipped on it, moreover it is one of the most popular personal belongings, we conclude that a smart phone is the most ideal device on which indoor location based service systems should be executed. Therefore, we propose a method of indoor positioning that determines the user's current position with the smart phone's sensor values. Referring to the accelerometer values, our method counts steps. With the orientation values, our method estimates the direction. With the step count and the direction, our method determines the current location of the user. Referring to the floor map, our method adjusts the user's current location from time to time. We have implemented the proposed method on Android smart phones and our test results are discussed in this paper.

Keywords: *Smart phone, Sensors, Indoor Positioning, Android App, Moving Status, Watching Status*

1. Introduction

Location of a user is one of the most important factors that determines the context of the user. Context-awareness has gotten attentions of software developers for a long time [1-6], for development of autonomously acting computing applications, and its role in teaching and learning is emphasized in [7].

Mobile phones are considered to be best suited for context-aware computing platforms because the owner of a mobile phone stores private information on the phone and keeps it on almost always. Among the mobile phones, smart phones are programmable and particularly tempting platform for building context-aware applications [8].

Context-awareness cannot be achieved without considering the user's location. In fact, location information alone can do so much for developing useful services that Location-based services (LBS) is one of the most prosperous industries.

People spend most of their time indoors. As the man-made constructions are getting bigger and bigger, demand for indoor Location-based services (LBS) is getting increased. However, most of the current LBS are for outdoor users and they do not work inside of buildings because GPS (Global Positioning System) is not available or operates poorly in indoor environments [9].

ISSN: 1975-4094 IJSH

Reliable, easy-to-use, and accurate indoor positioning is a key enabler of indoor LBSs, just as GPS has enabled outdoor LBSs. Examples of indoor LBSs include museum tour guides, and boarding reminders to air passengers far away from their gate, to name only two [10].

The WLAN-based indoor positioning is widely used because WLAN is available virtually everywhere. However, its performance is poor because of its Ad hoc layout and signal fluctuation [11]. Since a smartphone is equipped with a computer and many pretty accurate sensors, we propose a dead reckoning indoor positioning app running on Android smartphones. It estimates moving distance and heading direction with the sensor values and the current location with the estimated distance and direction. It also adjusts the estimated current location with information represented on the floor map. Our test results showed that our indoor positioning app is very accurate.

2. Related Works

It is obvious that context-aware applications are much more useful than non-context-aware ones and development of context-aware application has been hot research topic for a long time. However, commercial context-aware applications are still rare. One of the reasons for the rareness of commercial ones is that developing a context-aware application is difficult and inefficient. In order to address this problem, the authors of [8] designed and developed a software platform for context-aware mobile applications called ContextPhone. ContextPhone consists of four modules: Sensors gathering context data from different sources, Communications connecting to external services, Customizable applications augmenting built-in applications, and System services launching background services. These modules are provided as a set of open source C++ libraries and source code components so that app developers can use them. As another approach to address the problem, the authors of [1], introduced a conceptual framework and software infrastructure with which a developer can rapidly develop prototypes of context-aware applications and fine-tune them.

The author of [7] proposed a context-aware Japanese polite expressions learning system. This system is context-aware because it determines user's context with user's profile, RFID tag and GPS, and so on. This paper also proposes another learning system that identifies the objects with RFID tags attached around the user and provides the user with the educational information.

The pioneers of indoor positioning are [12-14]. In the Active Badge System [12], a user wears a badge that emits a unique infrared signal. Infrared sensors placed in the building pick up these signals. Then the system determines the user's current location with these signals.

In the Active Bat system [13], the Active bat tag attached to the user emits an ultrasonic pulse. The system estimates the distances between the user and the receivers with the ultrasound time-of-flight. The system calculates the user's location with the distances and the coordinates of the receivers.

In Cricket [14], a small device attached to a mobile user receives both RF and ultrasound signals sent by a beacon and determines the distance to the beacon. With the distance, the device attached to the user estimates the user's location. Therefore, user's location is not known to the system.

There are many references that are closely related to our work. The authors of [15] proposed a new smartphone-based indoor positioning system. The system determines user's moving direction with both magnetometer and gyroscope measurements. The system estimates the number of steps taken by the user with accelerometer values. The number of steps yields the distance the user moved. With the direction and the distance, the system estimates the user's current location with the following equation, where s_k , l_k , θ_k , s_E and

s_N represent the current user position, step length, the estimated heading orientation, and the East and the North coordinates, respectively.

$$s_{u,k} = \begin{bmatrix} s_{E,k} \\ s_{N,k} \end{bmatrix} = \begin{bmatrix} s_{E,k-1} + l_k \bullet \sin(2\pi\theta_k) \\ s_{N,k-1} + l_k \bullet \cos(2\pi\theta_k) \end{bmatrix}$$

In [15], they use gyrometer on the smartphone to obtain Azimuth, Pitch and Roll. With Azimuth, we can identify which direction the user is heading:

$$Direction = \begin{cases} N, \textit{if}\ 315 < Azimuth \le 45 \\ E, \textit{if}\ 45 < Azimuth \le 135 \\ S, \textit{if}\ 135 < Azimuth \le 225 \\ W, \textit{if}\ 225 < Azimuth \le 315 \end{cases}$$

The authors of [16] used the finite state machine (FSM) shown in Figure 1 in order to count steps. States S_0 and S_1 represent not walking and possibly started a step, S_2 and S_3 represent positive and negative peek has been reached, S_4 and S_5 are used to tolerate the noise, and S_6 represents the terminal state. The values they used for the variables are:

Thr: 0.6, Pos_Peek_Thr: 1.8, Neg_Peek_Thr: -1, Neg_Thr: -0.6

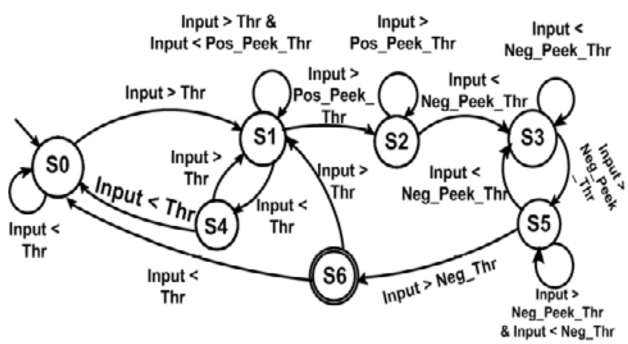

Figure 1. The Finite State Machine (FSM) Introduced in [11]

The authors of [17] also introduced a step-counting method. They collect accelerometer value periodically for a certain period of time, calculate the average and standard deviation of those collected values. With the average and the standard deviation, they calculate u_Lim and D_Limk as follows:

u_Lim = Average + Standard Deviation

d_Lim = Average + Standard Deviation.

They increase the number of steps whenever one or more sensor values greater than u_Lim are followed by one or more sensor values less than d_Lim.

3. Design of our System

Our system recognizes moving status, watching status, and the current location of the user. Our system collects sensor values every 50 millisecond. It calculates the standard deviation of recent 20 y axis accelerometer values among the collected sensor values. If the calculated standard deviation is less than a certain threshold then it determines that the user is not moving.

There is an obvious flaw in the FSM in Figure 1 at state S_2. That is if input is greater than Positive_Peek_Thr then it stays there, whereas if input is less than Negative_Peek_Thr then it moves to S_3. The question is what if input is greater than Negative_Peek_Thr and less than Positive_Peek_Thr. Therefore, for our indoor positioning, we use the modified FSM as shown in Figure 2.

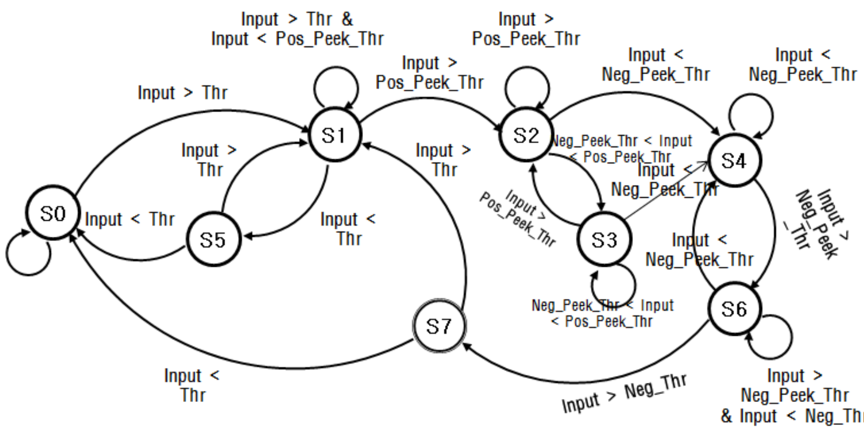

Figure 2. Our Finite State Machine to Count Steps

From S_2 if input is greater than Negative_Peek_Thr and less than Positive_Peek_Thr then the next state will be S_3. From S_3, if input is greater than Negative_Peek_Thr and less than Positive_Peek_Thr then we stay there, if input is greater than Positive_Peek_Thr then the next state is S_2, otherwise (input < Neg_Peek_Thr) the next state is S_4.

The final context factor that our system detects is watching status. If a user does not move and holds the Android smartphone in the portrait orientation then we determine that the user is watching an object (exhibit).

Our system periodically (every 50 milliseconds) collects sensor values. For collecting sensor values, we run a thread. Our system also checks watching status with recently collected 20 sets of sensor values (collected for 1 second). If it is determined to be watching status, our system applies the dead reckoning process on the sequential sets of sensor values collected from the last watching status. This dead reckoning process applies the process of counting steps shown in Figure 2 and produces a location at every step. Then, it displays the sequence of locations on the screen. A floor map provides the configuration of the floor including the locations of all exhibits. Using the floor map, our system adjusts the current location of the user.

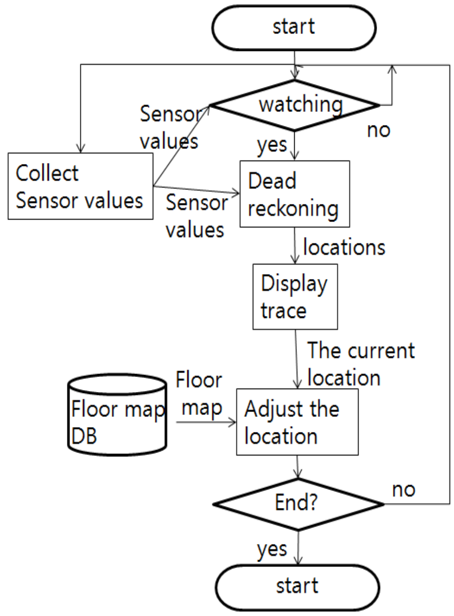

Figure 3. A Flow Diagram Describing the Process of our System

4. Implementation of our System

If the standard deviation of recently collected 20 y axis accelerometer values is less than a certain threshold, then our system determines that the user is not moving. We have collected y axis accelerometer values in various situations and an example collection is shown in Figure 4. By investing the collected data, we found that it 100 % correctly detects not moving status when the threshold is 0.5.

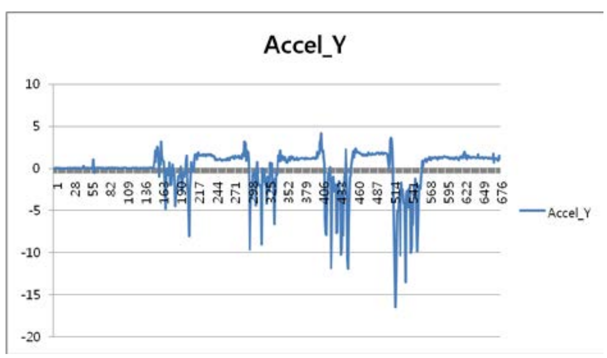

Figure 4. An Example Collection of Y-axis Accelerometer

Among the collected sensor values, we use the pitch value in order to determine the position of holding the smart phone. The pitch value is the degree/angle of the elevation of the line-of-sight determined by the camera and the object. When the value of the pitch (variable values [1]) is close to -90 and the user is not moving, we determine that the user is watching an exhibit.

Our system counts steps by investigating z axis accelerometer. An example of collected values of z axis accelerometer is shown in Figure 5. From the collected values, we found the proper thresholds as shown in Table 1.

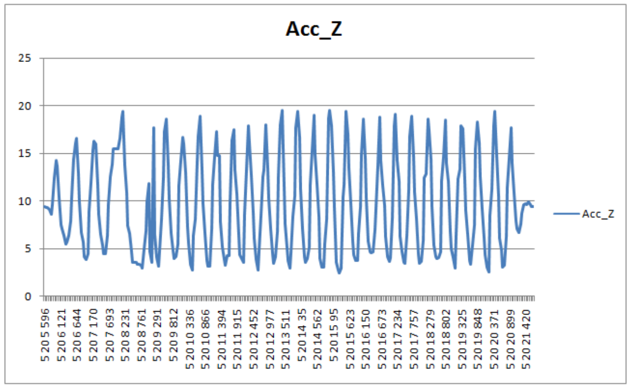

Figure 5. An Example Collection of Z Axis Accelerometer

Table 1. Font Sizes of Headings. Table Captions should always be Positioned above the Tables

Variables	Neg_Peek_Thr	Neg_Thr	Thr	Pos_Peek_Thr
Values	8.0	9.0	10.0	11.0

5. Experiments

We have implemented our system on an Android smart phone and performed experiments of testing the app in a virtual exhibition room. The configuration of the room is shown in Figure 6. The coordinates of the entrance is (6100, 4120) and the unit is mm. We have to keep it in the mind that the y axis coordinate grows as the point moves down not up. There are 11 exhibits displayed in the room. For each of the exhibits there is a proper position to watch it and the points are numbered from 1 to 11 in the figure. The coordinates of those points are (4350, 4500), (1950, 4500), (2000, 4700), (2000, 7500) and so on. We obtained the coordinates by measuring the distances and angles from the entrance to the points. For example, the y coordinates of points 1 and 2 are identical because the line defined by these two points is parallel to the horizontal line of the figure.

In order to check if the horizontal line of the figure coincides with the horizontal line of the earth, we have collected the orientation values while we walked from point 2 to point 1. A typical result of the test is shown in Figure 7. After many tests, we concluded that the angle defined by the horizontal line of the figure and that of the earth is about 20 degrees (We will call this angle theta from now on).

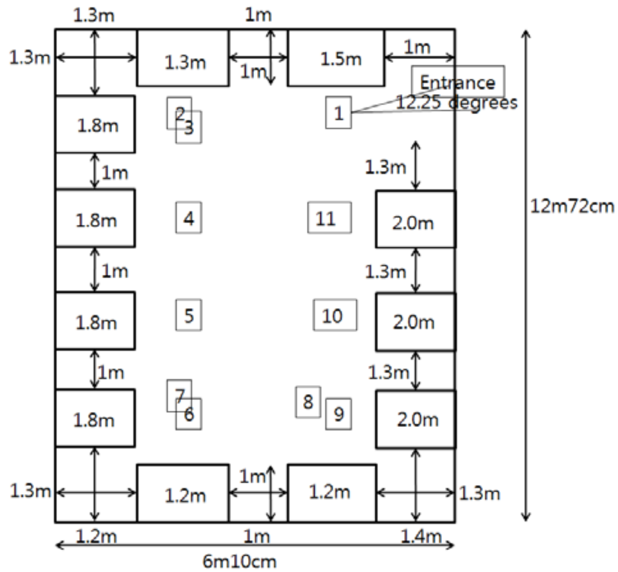

Figure 6. The Configuration of the Virtual Exhibition Room

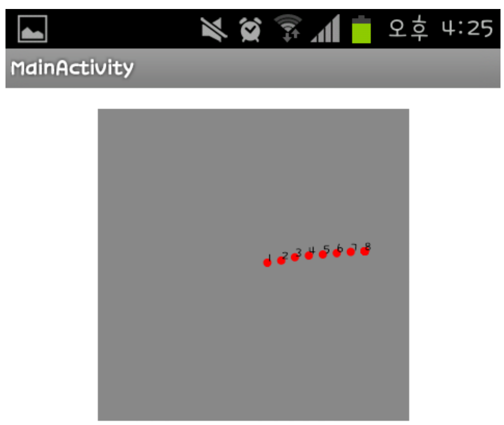

Figure 7. A Trace We Obtained while we Walk from West to East

In order to test our indoor positioning that returns window coordinates of the current position, we have to translate the coordinates on the figure into the coordinates of a smart phone. We determined the scale of the window over the exhibition room be 1/50 and to put the entrance point at (250, 50). The distance from the entrance to point 1 is 1790.78 mm and the angle determined by the horizontal line of the figure and the line determined by point 1 and the entrance point is 12.25 degrees. Therefore, we can obtain the window coordinates of point 1 as follows:

x coordinate of Point 1 = 250 -((Cos(Radian(theta+12.25))*1790.78)/50) = 219.7

y coordinate of Point 1 = 50 + ((Sin(Radian(theta+12.25))*1790.78)/50)=69.11.

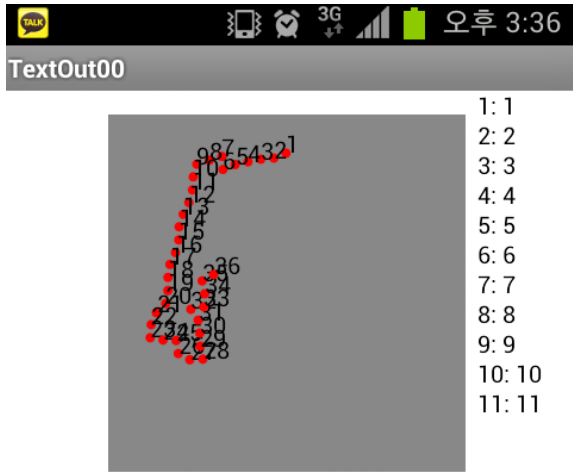

Figure 8. An Example Test Result of our Indoor Positioning Process

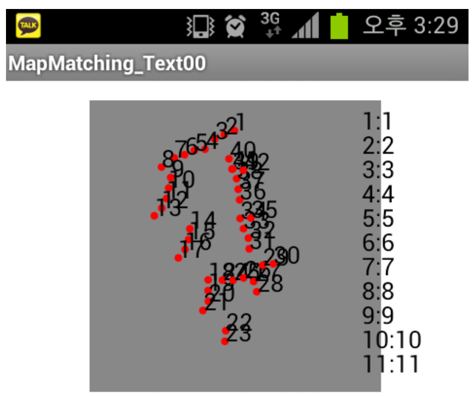

Figure 9. An Example Test Result of our Android App

We tested our positioning app while walking around the exhibition room. A typical example of our test results is shown in Figure 8. In this test, we started from the entrance and visited the exhibits from 1 to 11 in that order. A dot on the figure represents the user's position when the app detects a step. The figure shows that the user took 36 steps. The distances from point 1 to 2 is 2.4 meters, from point 3 to 6 is about 6 meters and the total distance the user walked is about 18.4 meters. In the positioning system, we assumed that the user moves very slowly looking around and the stride was about 0.5 meter. Therefore, we can conclude that our counting steps program is almost 100 % correct. Next to the window, we can see two columns of numbers. The left column represents the number of detected 'not moving' status. Since the user actually stopped 11 times, we can conclude that our system detects 'not moving' status 100 % correctly. The right column represents the exhibit our system determined that the user is watching at the stop and it shows 100 % correctly.

Finally, our indoor positioning Android application adjusts the result location obtained from the positioning process to the nearest watching position. An example test results of our Android app is shown in Figure 9. As we can see, the trace is not smooth. For example, the location of step 14 is far away from that of step 13. At the 14th step, the app detected that the user is at watching status. Then, it finds that the user is facing west by investigating the orientation value. Finally, it finds that the nearest watching location from the estimated location is exhibit 6 that is (2000, 13100) and it assigns it to the current location. From the figure, we can conclude that our Android app 100 % correctly recognizes the visited exhibits even if the room is very big.

6. Conclusions

We have introduced an Android app that detects the exhibit a visitor of an exhibition room is watching. For this, it detects the user's current location, moving status, and watching status. This paper described how the app makes use of the sensor values in order to detect those things. We can safely assume that the floor map of every exhibition room is available. For each of the exhibits, there are one or more positions from which a user can watch it most conveniently. This paper shows that our app will work perfectly even if the exhibition room is huge, using information on the floor map. For the future work, we are developing more valuable Android application on top of this application.

Acknowledgements

This work (Grants No. C0033172 and 2012-0002611) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012 and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology.

References

[1] K. Henricksen and J. Indulska, "A Software Engineering Framework for Context-Aware Pervasive Computing", Second IEEE Annual Conference on Pervasive Computing and Communications, **(2004)**, pp. 77-86.
[2] H. Lee and J. Kwon, "Combining Context-Awareness with Wearable Computing for Emotion-based Contents Service", IJAST, vol. 22, **(2010)**, pp. 13-24.
[3] S. Tsang and S. Clarke, "Mining User Models for Effective Adaptation of Context-aware Applications", IJSIA, vol. 2, no. 1, **(2008)**, pp. 53-62.
[4] S. Qi, M. Xi and Y. Qi, "Isotope Programming Model for Context aware Application", IJSEIA, vol. 1, no. 1, **(2007)**, pp. 53-66.
[5] J. Song and S. Kim, "Community based Context-aware Information for the Intelligent Personalized Information Service", IJMUE, vol. 3, no. 3, **(2008)**, pp. 23-30.
[6] J. Hsu, W. Lain, C. Chan and Y. Huang, "Implementation of IMS-based PoC Service with Context-Aware Interaction", IJMUE, vol. 3, no. 4, **(2008)**.
[7] H. Ogata and Y. Yano, "Context-Aware Support for Computer-Supported Ubiquitous Learning", 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04), pp. 27-34, **(2004)**.
[8] M. Raento, A. Oulasvirta, R. Petit and H. Toivonen, "ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications". IEEE Pervasive Computing, vol. 4, no. 2, **(2005)**, pp. 51-59.
[9] W. Kang, S. Nam, Y. Han and S. Lee, "Improved Heading Estimation for Smartphone-Based Indoor Positioning Systems", 23rd IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), **(2012)**, pp. 2449-2453.
[10] A. Baniukevic, D. Sabonis, C. Jensen and H. Lu, "Improving Wi-Fi Based Indoor Positioning Using Bluetooth Add-Ons", 12th IEEE International Conference on Mobile Data Management (MDM), **(2011)**, pp. 246-255.
[11] J. Zhou, W. Yeung and J. Ng, "Enhancing Indoor Positioning Accuracy by Utilizing Signals from Both the Mobile Phone Network and the Wireless Local Area Network", International Conference on Advanced Information Networking and Applications, **(2008)**, pp. 138-145.

[12] A. Hopper, A. Harter and T. Blackie, "The Active Badge System", INTERCHI-93, Amsterdam, The Netherlands, **(1993)**, pp. 533-534.
[13] A. Ward, A. Jones and A. Hopper, "A New Location Technique for the Active Office", IEEE Personnel Communications, vol. 4, no. 5, **(1997)**, pp. 42-47.
[14] B. Nissanka, A. Priyantha and H. Balakrishnan, "The Cricket Location-Support System", 6th ACM International Conference on Mobile Computing and Networking (ACM MOBICOM 2000), Boston, MA **(2000)**, pp. 32-43.
[15] . Leu and H. Tzeng, "Received Signal Strength Fingerprint and Footprint Assisted Indoor Positioning Based on Ambient Wi-Fi Signals", 75th IEEE Vehicular Technology Conference (VTC Spring), IEEE Press, New York, **(2012)**, pp. 1-5.
[16] M. Alzantot and M. Youssef, "UPTIME: Ubiquitous Pedestrian Tracking Using Mobile Phones", IEEE Wireless Communications and Networking Conference, IEEE Press, New York, **(2012)**, pp. 3204-3209.
[17] Y. Chon and H. Cha, "LifeMap: A Smartphone-Based Context Provider for Location-Based Service", IEEE Pervasive Computing, vol. 10, no. 2, **(2011)**, pp. 58-67.

Author

Jaegeol Yim received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.

