Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 540512, 10 pages
http://dx.doi.org/10.1155/2015/540512

Research Article

Adaptive Power Saving Method for Mobile Walking Guidance Device Using Motion Context

Jin-Hee Lee, Yeong-Ju Lee, Minseok Song, and Byeong-Seok Shin

Department of Computer Science and Information Engineering, Inha University, Incheon 22212, Republic of Korea

Correspondence should be addressed to Byeong-Seok Shin; bsshin@inha.ac.kr

Received 17 August 2015; Revised 9 November 2015; Accepted 10 November 2015

Academic Editor: Javid Taheri

It is important to recognize the motion of the user and the surrounding environment with multiple sensors. We developed a guidance system based on mobile device for visually impaired person that helps the user to walk safely to the destination in the previous study. However, a mobile device having multiple sensors spends more power when the sensors are activated simultaneously and continuously. We propose a method for reducing the power consumption of a mobile device by considering the motion context of the user. We analyze and classify the user's motion accurately by means of a decision tree and HMM (Hidden Markov Model) that exploit the data from a triaxial accelerometer sensor and a tilt sensor. We can reduce battery power consumption by controlling the number of active ultrasonic sensors and the frame rate of the camera used to acquire spatial context around the user. This helps us to extend the operating time of the device and reduce the weight of the device's built-in battery.

1. Introduction

Recently, the mobile devices are equipped with a variety of sensors, such as a GPS receiver, an accelerometer, a gyro sensor, and a camera, for recognizing the user's motion and environment. Efficient utilization of these sensors has therefore been studied [1–3]. However, one of the difficult issues is the residual time of battery in the mobile device when it activates several sensors continuously. Some of the sensors in the mobile, such as the camera, spend lots of battery power. Therefore, the power saving method for effectively using the sensors is required.

Another issue is the difficulty of extracting precise data from the sensors in the mobile device. Accelerometers and tilt sensors in particular are used to detect the motion context, which means relationships between the motions of the user during a certain period of time. In addition, it involves motion scale analysis and direction of the user's motion. However, detecting the exact motion is not easy because the data extracted from the sensors can be noisy and determining the motion features such as deviation and mean is difficult.

We propose the method to detect motion of the user by extracting more accurate data and to save the power by activating sensors efficiently. In order to reduce the operating frequency of the sensors consuming a lot of power, we activate the sensors only if you need to use sensors by analyzing the user's motion accurately. We determine the motion of the user by analyzing the data gathered from the accelerometer and the tilt sensor, which are low power consumption and low price compared to others. This method enables us to control the operation of other sensors adaptively. We can thus prolong the operating time of the mobile device and/or decrease the weight and size of its battery. In order to verify the availability of our proposed method, we applied it to a guidance system for visually impaired person that was developed in our previous studies [4]. It is based on mobile device and is used of additional sensors to detect the surroundings. We use a camera to estimate indoor position of the user and multiple ultrasonic sensors to avoid obstacles on the path. The device can save the power consumption about 15% by adjusting the frequency of use of the sensor in accordance with the user's motion, as compared with the case of activating the sensors consistently. In addition, the system not using the display device can save the power of about 40% in comparison with the case of activating the display device continuously.

Our method utilizes the sensor data processing technique so as to improve the recognition rate and accuracy despite the dynamic movement of a user possessing the mobile device. In addition, the motion recognition accuracy of our method is higher than that of the previous methods which use the data acquired from sensors attached to some parts of the user's body. The method detects user's motion with about 90% accuracy because of using specific features such as vertical and horizontal components and applying an HMM-based classifier to improve performance. As a result, the method could precisely detect motion of the user and effectively reduce the power consumption of the system.

We summarize related work in Section 2. In Section 3, we describe our method in detail. In Section 4, we present the experimental results applying the proposed method. We conclude our study in Section 5.

2. Related Work

In general, motion context refers to the activity pattern of a user as analyzed using the data extracted from sensors attached to some parts of the user's body. Kern et al. [5], Krause et al. [6], Ravi et al. [7], Choudhury et al. [8], and Karantonis et al. [9] researched human activity and context awareness using several accelerometer sensors. They analyzed the motion of the user with only data of accelerometer sensors. Those methods have no orientation problem for collecting data by attaching a sensor to a specific location on the body. However, our method reveals the orientation problem because it collects direction data from a mobile device. It is necessary to extract orientation-independent features that reflect the current position of the device, regardless of the orientation of the mobile device. A solution to avoid orientation problem is using magnitude of the accelerometer's each axis. Mizell [10] has shown that the average on each axis over a time period can produce an estimation value of the gravity-related component. We use a similar approach to estimate the gravity component from each axis of accelerometer sensor.

In the analysis the data of accelerometer sensors, methods for identifying user motion generally use a classifier, such as a decision tree and a GMM (Gaussian Mixture Model). Huynh and Schiele [11] categorized activities such as walking, writing, or sitting using an SVM (Support Vector Machine), and Long et al. [12] used a decision tree to classify a variety of human motions. Husz et al. [13] applied an APM (Action Primitive Model) that analyzed the motion using supervised learning, and Nakata [14] classified the activities by means of an approximate HMM (Hidden Markov Model). Zhu and Sheng [15] used an HMM for analyzing motion data extracted from accelerometer sensors attached to the hand or foot. However, methods using classifiers require additional processing to improve the accuracy and much training data to yield correct classification.

Mobility is an important factor for mobile devices because the power is continuously supplied from their battery. The device must be usable for long periods using a battery of small capacity. To reduce the power consumption of these devices, several methods have been devised to minimize the use of the CPU and the display [16–18]. The methods use other systemic energy optimization techniques so that the overall battery life of the device is increased [19–21]. However, they have problems that the response time of the device is delayed and the performance is degraded.

In this paper, we exploit the accelerometer sensor and tilt sensor embedded in the mobile device simultaneously to evaluate motion of the user and apply a decision tree based on approximation HMM for accurate analysis of the motion in real time. The proposed method can reduce power consumption because it minimizes CPU computations by controlling the frame rate of a camera and the number of active ultrasonic sensors used for recognizing the context of the user's surroundings, without loss of performance such as the processing speed. In other words, the method can save the power by adjusting sensors adaptively in a mobile device based on the motion recognition of the user.

3. Power Consumption Control Method

We propose a method for reducing power consumption by adjusting the frequency of the use of active sensors applied for context awareness. The proposed method consisted of two stages, motion analysis and power control. First, the method takes advantage of the motion context of the user derived from the accelerometer sensor and the tilt sensor. The motion of the user is analyzed in terms of the acceleration data for X, Y, and Z axes obtained from a triaxial accelerometer sensor. In other words, the motion analysis is conducted with some features such as vertical or horizontal acceleration components of user's action. In addition, we use the tilt sensor to correct errors in the data generated in accordance with the mounted position of the accelerometer sensor and the walking style of the user. To analyze accurately the motion context from both sensors, we apply an HMM-based decision tree which is a classification technique applying the time series method.

Depending on the result of this motion analysis, we determine the frequency of use of active sensors, which consume a lot of power in the system. It is to determine the minimum number of ultrasonic sensors required to be active and the minimum frame rate for the camera. Then, the recognition accuracy in that case should be similar to the accuracy in the case when using all sensors. By activating the necessary sensors only in special situations, rather than activating all sensors continuously, it is possible to reduce power consumption and to extend the battery life of the device.

We present an overview of the proposed method in Figure 1. It comprises two stages, namely, *analyzing* the motion context with the HMM and *controlling* the power consumption according to the identified situation, via the activation of specified sensors only.

3.1. Motion Context Estimation Using Accelerometer Sensor.

We analyze the motion and orientation of the user by means of mobile device's built-in triaxial accelerometer and tilt sensor. However, it is not easy to detect the motion directly from those sensors' data. Accurate motion recognition is

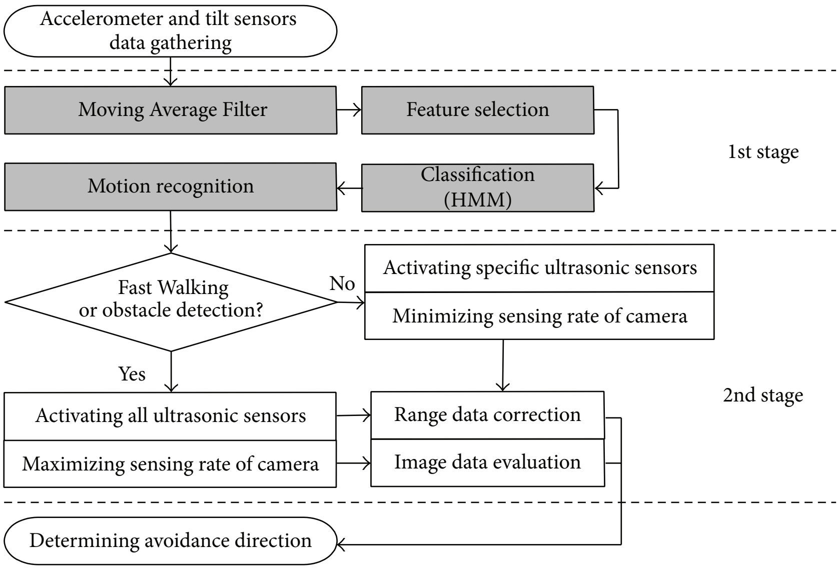
Accelerometer and tilt sensors data gathering
Moving Average Filter
Feature selection
1st stage
Motion recognition
Classification (HMM)
Fast Walking or obstacle detection?
No
Activating specific ultrasonic sensors
Minimizing sensing rate of camera
Yes
2nd stage
Activating all ultrasonic sensors
Maximizing sensing rate of camera
Range data correction
Image data evaluation
Determining avoidance direction

Figure 1: A flowchart of our proposed method.

difficult because some of the data may be lost or may contain noise [22, 23]. We therefore use probabilistic inference to construct a Weka Toolkit based decision tree using an HMM classifier that exploits both current data and previous data [24, 25]. We can analyze a variety of motions with the data extracted from the sensors. However, we focus on five motions such as *Standing*, *Walking*, *Fast Walking*, *Ascending Stairs*, and *Descending Stairs*. In addition, we want to choose three motions (*Standing, Walking, and Fast Walking*) based on the walking speed of the user. Three motions require significantly different amounts of power to activate sensors needed for context awareness [26, 27].

We acquire acceleration data for the device in the X-axis, Y-axis, and Z-axis directions from the triaxial accelerometer sensor. However, the data are erroneous because of jittering noise, even if the device has been placed on a table. To reduce the jittering noise, we scale down the acceleration data by applying an MAF (Moving Average Filter), as given by (1). Here, x_1, y_1, and z_1 are the raw data and x_2, y_2, and z_2 are the scaled-down data. The factor k defines the number of data according to the sampling time interval, and S indicates the span value for smoothing. This smoothing technique for noise reduction can be applied to both mobile and stationary devices

$$(x_2, y_2, z_2) = \mathrm{MAF}\left(\mathrm{round}\left[\frac{(x_1, y_1, z_1)}{k}\right], S\right). \tag{1}$$

Orientation problems may occur, because every person has a different gait and the mounted position of the accelerometer sensor is variable [7, 13]. To solve this problem, we use the magnitude values from the sensor as well as the orientation-independent features such as the standard deviation and the mean. These are obtained from the vertical and horizontal components of accelerometer sensor. At this stage, we have determined the sampling period for calculating each value via repeated experiments. Let the acceleration vector at any point be V_{acc} and let the mean value for each axis be X_m, Y_m, and Z_m. We then define a reference vector $V_n = (X_n, Y_n, Z_n)$, which is normalized from V_{acc}. As described in (2), we derive the horizontal vector V_{hor} from V_{acc} and the vertical vector V_{ver}, which is multiplied by p_i and V_n. p_i means a scalar value being the inner product of V_{acc} and V_n

$$V_{\mathrm{hor}} = V_{\mathrm{acc}} - V_{\mathrm{ver}}, \tag{2}$$

where

$$\begin{aligned} p_i &= \left\langle V_{\mathrm{acc}}, V_n \right\rangle, \\ V_{\mathrm{ver}} &= p_i \cdot V_n. \end{aligned} \tag{3}$$

We evaluate the horizontal and vertical components by means of estimating horizontal and vertical vectors. A horizontal magnitude defines $|V_{\mathrm{hor}}|$ and a vertical magnitude uses V_{ver}. To determine the parameters used in the classifier, we estimate features such as mean, standard deviation, 75% percentile range, and zero crossing rate, computed from the waveform of magnitude. To gather sufficient training data, acceleration data are collected from test users over about four hours. Each person carries out the three motions (*Standing*, *Walking*, and *Fast Walking*). We use a C4.5 decision tree that is known to increase the recognition accuracy by increasing the number of samples [28]. The tree classifier involves the features of the motion, as the mean and standard deviation of the vertical and horizontal components of the acceleration. We define meanV and stdV as the vertical features and meanH and stdH as the horizontal features. We generate a well-pruned decision tree (shown in Figure 2) based on k-means clustering for matching similar motions. However, there is a limit to recognize two motions (*Walking* and *Fast Walking*) if using only decision tree because the motions show various changes of complicated patterns according to the time. Therefore, in order to improve the motion classification accuracy, we create sequence data by collecting

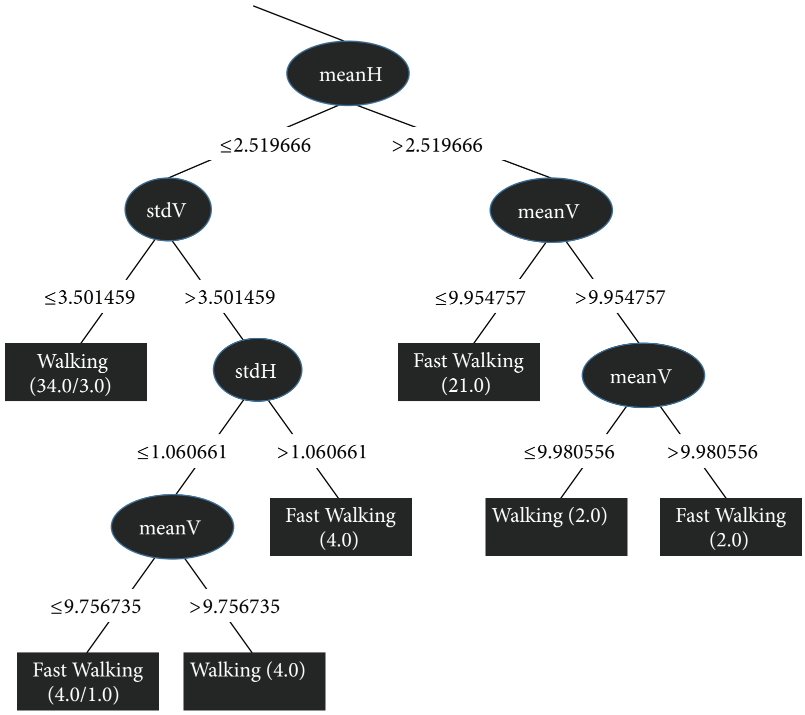

Figure 2: An example of generating a decision tree based on approximation HMM.

classification results of predetermined length obtained from a decision tree. Then, we use approximation HMM which is a classification technique applying the time series method. In other words, we employ the Viterbi algorithm based on an HMM to maximize the utilization of the correlation between continuous motions [29, 30].

3.2. Adaptive Power Control via Motion Context. To verify the effectiveness of the proposed method, we implemented a prototype system that acquired the user's spatial context using a variety of sensors. To reduce power consumption, we controlled the frame rate and the number of active sensors based on the motion context. The prototype system could detect an obstacle in the user's path via six ultrasonic sensors. To recognize objects in front of the user, it is important to arrange the sensors efficiently to cover the maximum range with the minimum number of sensors based on each sensor's physical characteristics, such as its coverage and the detection range. In addition, the sensors should detect obstacles quickly and precisely. Therefore, we estimate the geometric information for all sensors and determine their optimal placement via repeated experimentation [31]. As depicted in Figure 3, we simplify the spatial structure in front of the user by classifying it as one of several predefined patterns. We then determine an avoidance direction by evaluating the pattern to minimize the probability of collision with the obstacle. As shown in Figure 4, we set each sensor's direction and coverage to overlap as little as possible with those of neighboring sensors, by considering the walking speed of the user and the sensing rate of the sensor [32].

We consider the range data extracted from four ultrasonic sensors and represent the spatial information in terms of patterns in front of the user. The range data are classified into four cases: danger (less than 100 cm), warning (100~130 cm), adequate (130~200 cm), and unconcern (more than 200 cm). We can identify 256 (= 44) cases and can generate the corresponding range data from the four sensors in each case. All cases are stored in a table (see Table 1). Each number denotes one of the four cases, namely, 0 (danger), 1 (warning), 2 (adequate), and 3 (unconcern). The avoidance instructions are classified into some cases, namely, turn-left, turn-right, and forward. The avoidance direction for the obstacle can therefore be determined by referring to the table.

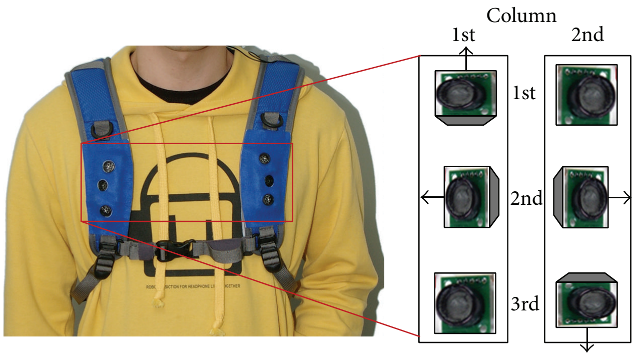

Figure 3: Layout of the ultrasonic sensors.

As shown in Figure 5(a), if the motion is recognized as *Walking* or the user proceeds straight ahead, we can deactivate the two sensors that sense spatial information to the left and the right of the user. This is because four sensors for detecting frontal space can detect obstacles placed to the left and the right of the user if the walking speed is average.

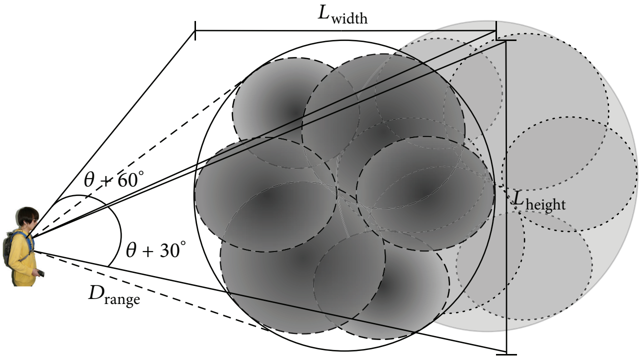

Figure 4: Scan coverage of the ultrasonic sensors.

Table 1: A decision table for determining the obstacle avoidance direction.

Case	Obstacle detection				Avoidance direction
	Left (L)	Forward (F_1)	Forward (F_2)	Right (R)	
0	0	0	0	0	Turn-right
1	0	0	0	1	Turn-right
⋮	⋮	⋮	⋮	⋮	⋮
64	1	0	0	0	Turn-left
65	1	0	0	1	Turn-right
66	1	0	0	2	Turn-right
⋮	⋮	⋮	⋮	⋮	⋮
254	3	3	3	2	Forward
255	3	3	3	3	Forward

It is therefore possible to reduce power consumption by selectively activating the sensors that are arranged in the same direction as the walking direction of the user. As shown in Figure 5(b), when the motion is perceived as *Fast Walking* or an obstacle is detected, we have to acquire spatial information to the left and the right of the user to avoid obstacles. In addition, it is necessary to analyze the frontal space precisely for *Fast Walking*. We therefore have to activate all ultrasonic sensors. This will enable us to detect an obstacle accurately even if the user walks fast.

In addition, we attached identifying markers to the ceiling at regular intervals to enable tracking of the position of the user via camera recognition of the markers. We increase the camera's frame rate for accurate recognition of the markers when the motion is recognized as *Fast Walking* and minimize the frame rate when the motion is perceived as *Walking*, as shown in Figure 6. The method can reduce the required battery power by decreasing the frame rate, while maintaining the detection accuracy, when the motion is recognized as *Walking*.

4. Experimental Results

4.1. Motion Patterns Analysis. It is very important to correctly classify the various human motions. We conducted experiments to compare the accuracy of several classifiers to detect specific motion from the input data, such as mean and standard deviation of the horizontal and vertical components obtained from the accelerometer sensor. We compare and analyze four classifiers: decision tree (DT), naïve Bayesian (NB), k-nearest neighbor (kNN), and logistic regression (LR) based on probabilistic inference techniques. A window size of the classifiers is set as 100 samples collected in the same duration for five motions: *Standing, Walking, Fast Walking, Ascending Stairs*, and *Descending Stairs*. Table 2 shows the accuracy of classification applying each classifier. As shown in the results, all the classifiers well sorted standing motion, but they showed lower accuracy for ascending stair motion in comparison to the other motions. A decision tree well classified all the motions compared to other classifiers. Therefore, we determine to use a decision tree as a motion classifier.

Table 2: Accuracy of motion recognition according to the classifier.

Classifier	Standing	Walking	Fast Walking	Ascending Stairs	Descending Stairs
DT	100	95.12	92.27	85.39	93.06
NB	99.14	77.59	82.37	85.25	87.93
kNN	89.93	63.33	72.36	61.38	50.23
LR	95.81	71.28	84.53	79.11	84.78

We construct a C4.5 decision tree, generated by the Weka Toolkit, which is known to be a relatively accurate method even with a small number of training samples [28]. We perform the training and execution phases of a process that detects motion. In the training phase, we collect users' motion. We calculate the mean and standard deviation of the horizontal and vertical components of the acceleration values continuously over a predefined period. We then generate the decision tree using N_s samples and N_a test data [4]. The accuracy of recognition increases with increasing N_a and N_s; however, in our experiments, we have obtained high accuracy even with small sample spaces. We identify three motions depending on the walking speed of the user: "*Standing* (0 km/h)," "*Walking* (less than 3 km/h)," and "*Fast Walking* (less than 5 km/h)." Also, the experiment includes results of *Ascending Stairs* and *Descending Stairs*. In the execution phase, the current motion is determined by exploring the decision tree. We can achieve accurate motion classification by periodically checking the horizontal and vertical components and by transferring only accurate values to the decision tree.

The size of the sample space is an important factor required for the decision tree. To determine a suitable value, we carried out experiments that measured the accuracy of the motion classification and the tree search time for various factor values. We collected a training data set and generated a decision tree having $3000N_s$ [33]. Table 3 reports the accuracy of motion detection and the tree search time for various values of N_a. The accuracy increases as the size of the N_a increases, but the classification computation time increases. In other words, the classification computation takes

Table 3: Motion detection accuracy and tree lookup time against N_a values.

N_a	Standing		Walking		Fast Walking		Ascending Stairs		Descending Stairs	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
20	84.33	0.10	74.25	0.09	37.75	0.09	21.74	0.08	29.12	0.09
30	89.21	0.18	80.75	0.16	63.33	0.13	50.58	0.14	50.29	0.17
50	99.50	0.25	87.58	0.23	83.50	0.23	74.39	0.25	87.71	0.27
80	99.89	0.40	92.76	0.37	90.28	0.35	83.26	0.39	91.58	0.38
100	100	0.53	95.12	0.48	92.27	0.46	85.39	0.47	93.06	0.49

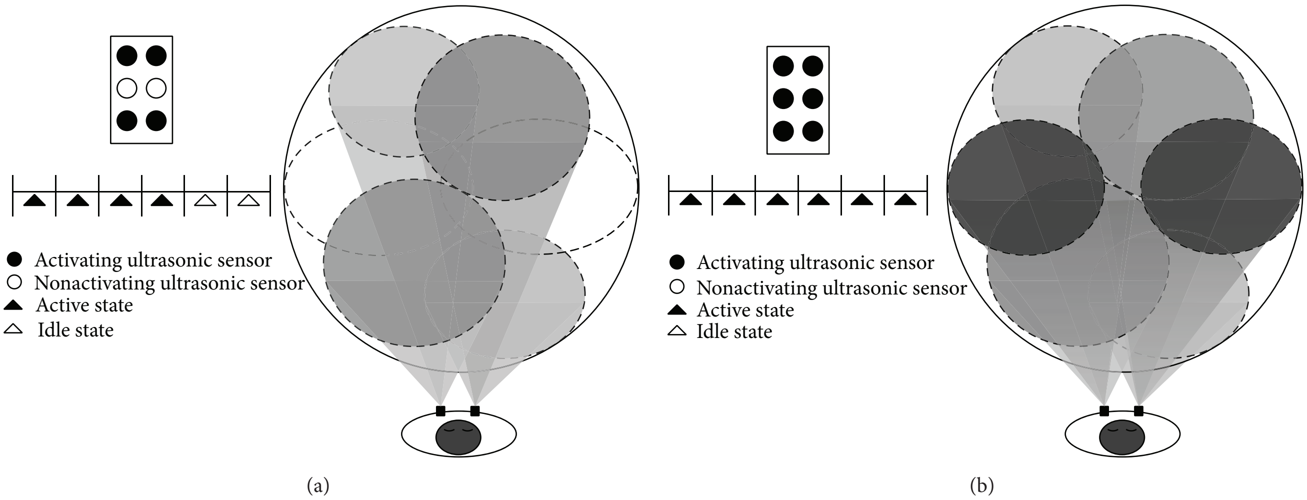
● Activating ultrasonic sensor
○ Nonactivating ultrasonic sensor
▲ Active state
△ Idle state
(a)
● Activating ultrasonic sensor
○ Nonactivating ultrasonic sensor
▲ Active state
△ Idle state
(b)

Figure 5: Detection range according to the status of sensors. (a) The motion is recognized as *Walking* or the user proceeds straight. (b) The motion is recognized as *Fast Walking* or the system detects obstacles.

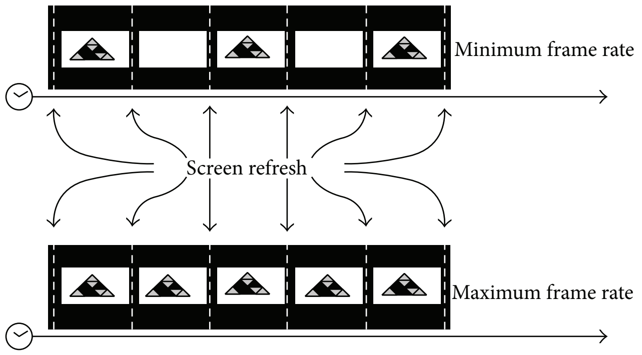

Figure 6: Change of the frame rate according to recognized motion.

more time if N_a is larger. In addition, the search time for recognizing the motion is proportional to N_a. Therefore, we design the tree by considering the trade-off between the accuracy of motion detection and the motion recognition time. From these experiments, we determined that N_a should be 50, because the results show that the detection accuracy of all motions is high and the computation is completed in 0.25 seconds, that is, the motion detected sufficiently accurately at the lowest cost.

We consider the number of active ultrasonic sensors and the sampling rate of the camera, which can be controlled according to three motions (*Standing*, *Walking*, and *Fast Walking*) requiring significantly different amounts of power. In case of the *Standing* state, we do not supply power to the ultrasonic sensor and the camera. When the motion is recognized as *Walking*, we activate only four ultrasonic sensors to detect obstacles in front of the user, and we capture the image as a frame rate of about 3 fps. When the motion is perceived as *Fast Walking*, we activate all ultrasonic sensors and operate the camera at its maximum frame rate (5 fps). Through a number of experiments, we determined the optimal number of active sensors and the sampling rate for the camera depending on the situation, aiming to maximize the accuracy of the motion detection and minimize power consumption. We constructed a confusion matrix from a decision tree using 10,000 samples. We present the results in Figure 7. We confirmed that the number of sensors and the frame rate of the camera changed adaptively according to the motion of the user, as shown in Figure 7.

4.2. Accuracy Measurement. We evaluated the performance with five randomly selected students aged between 20 and 40 and four visually impaired persons. The users were not familiar with the experiment and the students were blindfolded. The obstacle placed on the path was a box (about 20 cm wide). We determined the optimal marker size as 12 × 12 cm, considering the distance from the ceiling to the ground

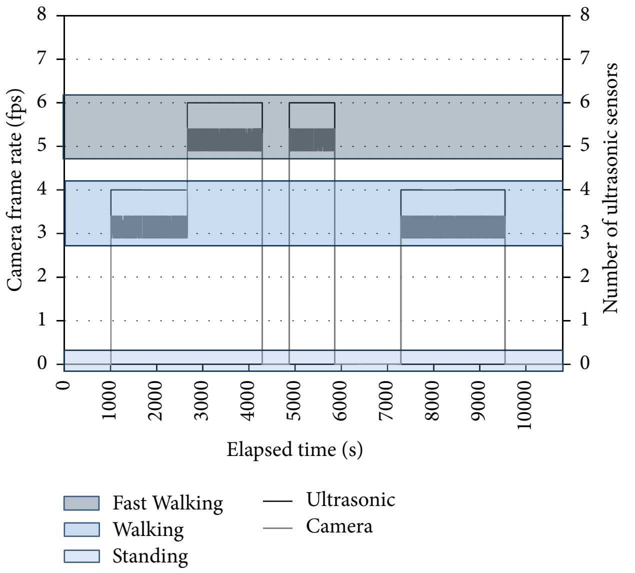

Figure 7: The change in the number of active ultrasonic sensors (right vertical axis) and frame rate of the camera (left vertical axis) according to the identified motion.

Table 4: Detection rate for the obstacle according to the number of ultrasonic sensors.

Case	Detection rate (%)		
	Standing (0 km/h)	Walking (≤3 km/h)	Fast Walking (≤5 km/h)
4 sensors	100	94	52
6 sensors	100	97	86

and the camera viewing angle. If the obstacle was detected, the user was required to walk until hearing the message "Stop." A scan using the six sensors required about 125 ms and the latency was set to 400 ms (the time between detecting an obstacle and providing feedback to the user). We determined that this latency offered sufficient time to react to any motion change by the user, via repeated experiments.

As shown in Table 4, we measured the detection rate for the obstacle for various numbers of sensors. When the motion was recognized as *Walking*, the detection rates were 94% and 97% for four and six active ultrasonic sensors, respectively. The experimental results were similar for both cases. However, for the case of *Fast Walking* at speed 67% faster than *Walking*, the detection rate for four active sensors is reduced by about 40% compared with six sensors. We therefore need to activate only four sensors (to reduce power consumption) during *Walking*. However, we should operate all sensors during *Fast Walking*, if we aim to maintain similar accuracy in both cases.

Figure 8 shows the rate of detection of the markers for different camera frame rates. We measured frame rates from 1 fps to 7 fps. However, we focused on three frame rates (2, 3, and 5 fps) that showed high accuracy and saved substantial power over repeated experimentation. The accuracy at 3 fps is similar to that at 5 fps if the motion is recognized as *Walking*.

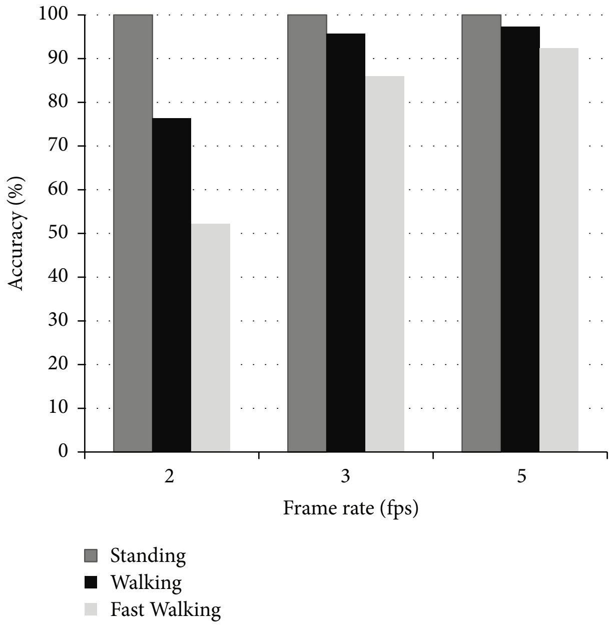

Figure 8: Detection rate of markers according to the frame rate of the camera.

However, the detection rate at 3 fps is substantially higher than that for the other frame rates. We therefore use 3 fps during *Walking* because it requires less power than the other frame rates, while offering similar accuracy.

4.3. Power Consumption Measurement. We measured the relative power consumption by setting a time slot (10,000 samples) and considering three patterns (activating six ultrasonic sensors, four sensors, and no sensors). Figure 9 shows the power consumption for various numbers of active ultrasonic sensors. We evaluated the power consumption from the current and voltage of the battery. The power consumed is equal to the product of voltage and current. Therefore, the power consumption is affected by the number of active sensors because of the sensor current. From the experimental results, about 450 mA was required if there were six active sensors, compared with 350 mA when none were activated. We can therefore reduce power consumption by controlling the number of active sensors based on the motion while maintaining the accuracy level required for obstacle detection.

Figure 10 shows the power consumed for various camera frame rates. We evaluated the relative power consumption by setting a time slot (10,000 samples) and considering three cases (sampling at 2 fps, 3 fps, and 5 fps). When sampling at 5 fps, about 17% more battery power is required than when sampling at 3 fps. As shown in the figure, the frame rate of the camera affects the current it requires and hence its power consumption. We can control the frame rate of the camera adaptively according to the user's motion to reduce power consumption while maintaining detection accuracy.

We measured the amount of power consumption in two cases: when and when not applying the motion context. The case applying the motion context is defined as SAS

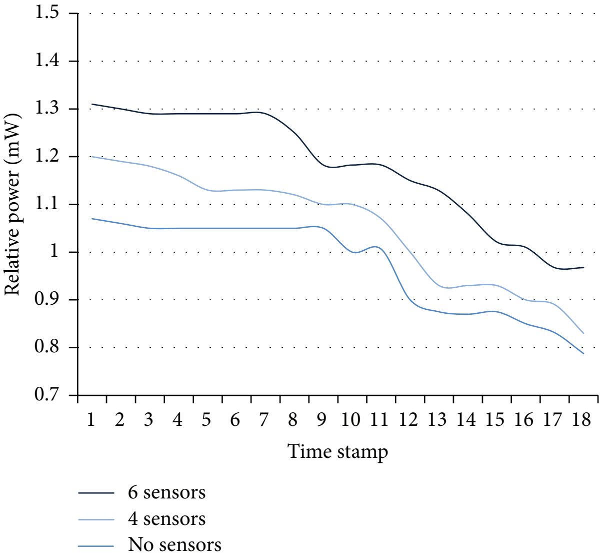

Figure 9: Relative power consumption for three cases: activating six sensors, four sensors, and no sensors.

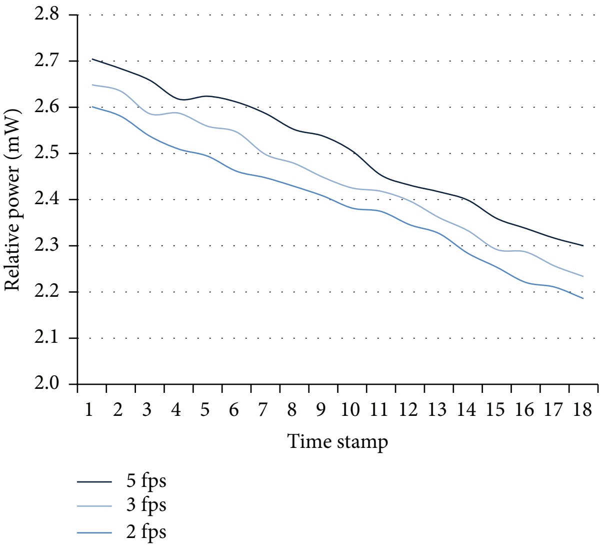

Figure 10: Relative power consumption for three cases: sampling at 2, 3, and 5 fps.

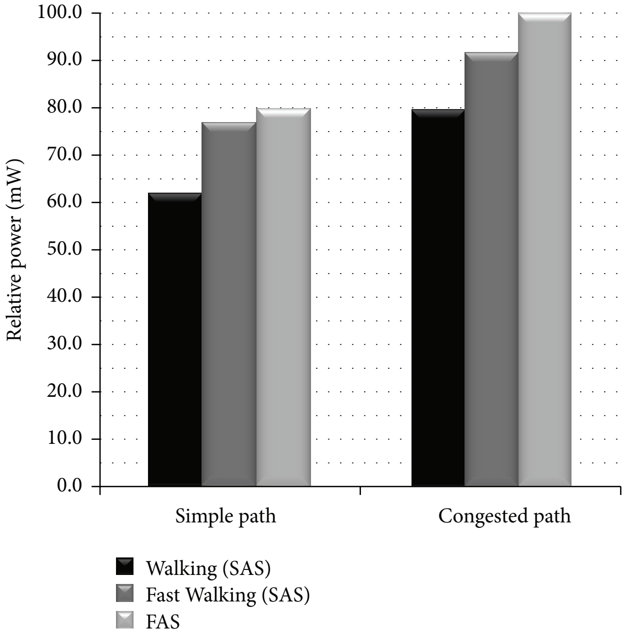

Figure 11: Comparison of power consumption of SAS and FAS system in accordance with two different road conditions.

(Selectively Activating Sensor) and the case not using the motion context is defined as FAS (Fully Activating Sensor). Each experiment was carried out in a simple path including obstacles and in a congested path having long walking distance. As shown in Figure 11, when the motion is recognized as *Walking* in a simple path, the system consumed less power about 15% than when it was recognized as *Fast Walking*. In addition, it showed a power reduction of about 18% compared to the case of FAS not using motion context. In a congested path, when the motion was perceived as *Walking*, the system spent less power about 12% than when it was recognized as *Fast Walking*. Furthermore, there was a power reduction effect of about 20% compared to the case of applied FAS. Therefore, we could verify the availability of the proposed method through experiments showing that there is a relative power saving of approximately 15% or more compared to that without using motion context.

5. Conclusions

In this paper, we analyzed the motion context of a user of a mobile device using data from its triaxial accelerometer and tilt sensor. We found that we could reduce the device's power consumption by controlling the number of active sensors and the frame rate of the camera used to acquire data about the spatial context, based on the user's identified motion. This enables the use of the device for an extended time and a reduction of the weight and size of the device, because it should be possible to reduce the capacity of the battery without excessively compromising performance. As future work, we are working on applying the proposed method in the smartwatch as one of the mobile devices. The proposed method can be applied in various mobile devices with 3-axis acceleration sensor and save the power by controlling the activation of sensors embedded on the mobile device.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (Grant no. HI14C0765). This work was supported by INHA University research grant.

References

[1] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN '06)*, pp. 114–116, IEEE, Cambridge, Mass, USA, April 2006.

[2] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, A. Ferscha and F. Mattern, Eds., pp. 1–17, Springer, Berlin, Germany, 2004.

[3] Y. Wang, J. Lin, M. Annavaram et al., "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 179–192, ACM, Kraków, Poland, June 2009.

[4] J. H. Lee, Y. Lee, E. S. Lee, J. S. Park, and B. S. Shin, "Adaptive power control of obstacle avoidance system using via motion context for visually impaired person," in *Proceedings of the International Conference on Cloud Computing and Social Networking (ICCCSN '12)*, pp. 1–4, Bandung, Indonesia, April 2012.

[5] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Ambient Intelligence*, vol. 2875 of *Lecture Notes in Computer Science*, pp. 220–232, Springer, Berlin, Germany, 2003.

[6] A. Krause, D. P. Siewiorek, A. Smailagic, and J. Farringdon, "Unsupervised, dynamic identification of physiological and activity context in wearable computing," in *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC '03)*, pp. 88–97, White Plains, NY, USA, October 2003.

[7] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence (IAAI '05)*, vol. 3, pp. 1541–1546, Pittsburgh, Pa, USA, July 2005.

[8] T. Choudhury, G. Borriello, S. Consolvo et al., "The mobile sensing platform: an embedded activity recognition system," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 32–41, 2008.

[9] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.

[10] D. Mizell, "Using gravity to estimate accelerometer orientation," in *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC '03)*, pp. 252–253, October 2003.

[11] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies (sOc-EUSAI '05)*, pp. 159–163, Grenoble, France, October 2005.

[12] X. Long, B. Yin, and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '09)*, pp. 6107–6110, IEEE, Minneapolis, Minn, USA, September 2009.

[13] Z. L. Husz, A. M. Wallace, and P. R. Green, "Human activity recognition with action primitives," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '07)*, pp. 330–335, London, UK, September 2007.

[14] T. Nakata, "Recognizing human activities in video by multi-resolutional optical flows," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 1793–1798, Beijing, China, October 2006.

[15] C. Zhu and W. Sheng, "Multi-sensor fusion for human daily activity recognition in robot-assisted living," in *Proceedings of the 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI '09)*, pp. 303–304, March 2009.

[16] M. Y. Malik, *Power Consumption Analysis of a Modern Smartphone*, Lecture Notes in Computer Science, 2012.

[17] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC Spring '11)*, pp. 1–6, Yokohama, Japan, May 2011.

[18] A. Shye, B. Scholbrock, and G. Memik, "Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42 '09)*, pp. 168–178, IEEE, New York, NY, USA, December 2009.

[19] M. Kennedy, A. Ksentini, Y. Hadjadj-Aoul, and G.-M. Muntean, "Adaptive energy optimization in multimedia-centric wireless devices: a survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 768–786, 2013.

[20] J. Yang and M. Song, "Threshold-based frame-rate adjustment for energy-saving in portable media players," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '15)*, pp. 50–51, Las Vegas, Nev, USA, January 2015.

[21] M. Kennedy, H. Venkataraman, and G.-M. Muntean, "Dynamic stream control for energy efficient video streaming," in *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB '11)*, pp. 1–6, Nuremberg, Germany, June 2011.

[22] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion context: a new representation for human action recognition," in *Computer Vision—ECCV 2008*, vol. 5305 of *Lecture Notes in Computer Science*, pp. 817–829, Springer, Berlin, Germany, 2008.

[23] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 261–272, Beijing, China, September 2009.

[24] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou, "Robust sequential data modeling using an outlier tolerant hidden markov model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1657–1669, 2009.

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[26] R. Cross, "Standing, walking, running, and jumping on a force plate," *American Journal of Physics*, vol. 67, no. 4, pp. 304–309, 1999.

[27] D. Winter, "Human balance and posture control during standing and walking," *Gait and Posture*, vol. 3, no. 4, pp. 193–214, 1995.

[28] A. Taherkhani, "Recognizing sorting algorithms with the C4.5 decision tree classifier," in *Proceedings of the 18th IEEE International Conference on Program Comprehension (ICPC '10)*, pp. 72–75, Braga, Portugal, July 2010.

[29] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of the 21st International Conference*

on Machine Learning (ICML '04), pp. 225–232, ACM, Alberta, Canada, July 2004.

[30] H.-L. Lou, "Implementing the Viterbi algorithm," *IEEE Signal Processing Magazine*, vol. 12, no. 5, pp. 42–52, 1995.

[31] J.-E. Kim, J. Han, and C.-G. Lee, "Optimal 3-coverage with minimum separation requirements for ubiquitous computing environments," *Mobile Networks and Applications*, vol. 14, no. 5, pp. 556–570, 2009.

[32] J.-H. Lee, E. Choi, S. Lim, and B.-S. Shin, "Wearable computer system reflecting spatial context," in *Proceedings of the 1st IEEE International Workshop on Semantic Computing and Applications (IWSCA '08)*, pp. 153–159, Incheon, Republic of Korea, July 2008.

[33] Y. Lee and M. Song, "Selective device activation for power reduction in accelerometer-based wearable guidance systems for the blind," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '12)*, pp. 106–107, Las Vegas, Nev, USA, January 2012.