Action Logger

Technical Report

Version 1.0

by

Mobileware Team

Ubiquitous Computing Lab

Kyung Hee University

Korea

Authors

Man Hyung Han (smiley@oslab.khu.ac.kr)
Tae Ho Hur (hth@oslab.khu.ac.kr)
Do Hyung Kim (dhkim@oslab.khu.ac.kr)
Won Ju Lee (wonju0618@oslab.khu.ac.kr)
Ha Neul Yeom (sky@oslab.khu.ac.kr)
Sang Beom Park (sbp@oslab.khu.ac.kr)

Supervised by

Professor Sungyoung Lee (sylee@oslab.khu.ac.kr)

<Table of contents>

<List of tables>

<List of figures>

Chapter 1 – Introduction

Activity recognition recognizes user's activity with the data acquired from sensors such as accelerometer, video, GPS, etc. Research of activity recognition has been conducted since 1980's and is still in progress. This has become a core technology in the field of next generation industries such as healthcare, lifelog and ubiquitous computing. Various research suggests diverse methods to recognize lots of activities with high accuracy[1].

Especially, research using accelerometer is one of the most popular way to recognize user's activity in the field of activity recognition. The research started assuming that there would be a change of intrinsic energy when a motion is changed, and use acceleration sensor to measure this change. Existing research tried to measure the change of motion precisely, so they attached lots of acceleration sensors in each part of the body and analyzed them[2,3].

But this kind of method is changing now after smartphone was released. On previous research, users have to buy a number of sensors with different types to attach them in each part of the body. This costs a lot that each sensor is expensive, and it is very cumbersome to be feasible in real life. By the emergence of smartphone, this trend is starting to change. According to statistics, one in every five people in the world own a smartphone now, and it will increase further[4]. As a smartphone is embedded with multimodal sensors, activity recognition research using smartphone is growing, and also is the accuracy.

Research using smartphone usually uses acceleration sensor, and shows high accuracy over 90% when recognizing physical activities such as stay, walking and jogging. But it shows low accuracy when recognizing bus and subway. This is because that physical activities like stay, walking and running have obvious difference of magnitude and pattern in acceleration signal, but not in vehicles like bus or subway. This means that it is vulnerable to noises which is lead by even subtle movements.

Activities such as stay, walking, jogging and taking a bus and subway are basic activities in daily life which happen frequently. Recognizing these activities is required for further studies, which is able to be used as a core technology in ubiquitous industry or lifelog service which needs a daily record to analyze daily life pattern of the user.

In this research, we propose a method to recognize daily activities using acceleration sensor and GPS from a smartphone. Our method guarantees high accuracy of recognizing stay, walking, jogging and even taking a bus and a subway irrelevant to the orientation and position of the smartphone. We use gyro sensor embedded in smartphone to correct the acceleration data and get fixed signal vector irrelevant to the direction of X, Y and Z axis. From among these, we use Z-axis to extract inherent vibration feature from bus and subway, and use these information to discriminate stay, bus and subway. We collected data and extracted features by carrying the smartphone in the pocket of top and bottoms, in the bag and holding it on hand which are the general positions the users carry the smartphone. And this led us showing a great accuracy while testing in the real field.

Chapter 2 – Related work

2.1 Acceleration sensor based activity recognition

Nishkam Ravi et al recognized daily activities based on one 3D acceleration sensor[3]. They experimented on various classifiers with same features to see which classifier shows the best accuracy, and see the difference of accuracy if the data collector and the experimenter is same or not. Eight activities were recognized such as standing, walking, running, climbing up an down the stairs, sit-ups, vacuuming and brushing teeth. The sensor was attached near pelvis, and total 12 features were used. They used FFT to each axis and calculated mean, standard deviation, energy and correlation. <Table 1> shows the result of the experiment.

<Table 1>Accuracy result of the experiment

Classifier	Accuracy(%)			
	Setting1	Setting2	Setting3	Setting4
Naive Bayes(NB)	98.86	96.69	89.96	64.00
Boosted NB	98.86	98.71	89.96	64.00
Bagged NB	98.58	96.88	90.39	59.33
SVM	98.15	98.16	68.78	63.00
Boosted SVM	99.43	98.16	67.90	**73.33**
Bagged SVM	98.15	98.53	68.78	60.00
kNN	98.15	99.26	72.93	49.67
Boosted kNN	99.15	99.26	72.93	49.67
Bagged kNN	99.15	99.26	70.52	46.67
Decision Table(DT)	92.45	91.91	55.68	46.33
Boosted DT	97.86	98.53	55.68	46.33
Bagged DT	93.30	94.85	55.90	46.67
Decision Tree(DTr)	97.29	98.53	77.95	57.00
Boosted DTr	98.15	98.35	77.95	57.00
Bagged DTr	97.29	95.22	78.82	63.33
Plurality Voting	**99.57**	**99.82**	**90.61**	65.33
Stacking (MDTs)	99.00	99.26	89.96	64.00
Stacking (ODTs)	98.86	98.35	84.50	64.00

Details of setting 1 to 4 in <Table 1> are shown below.

Setting 1 : Data Collected for a single subject over different days, mixed together and cross-validated.

Setting 2 : Data collected for multiple subjects over different days, mixed together and cross-validated.

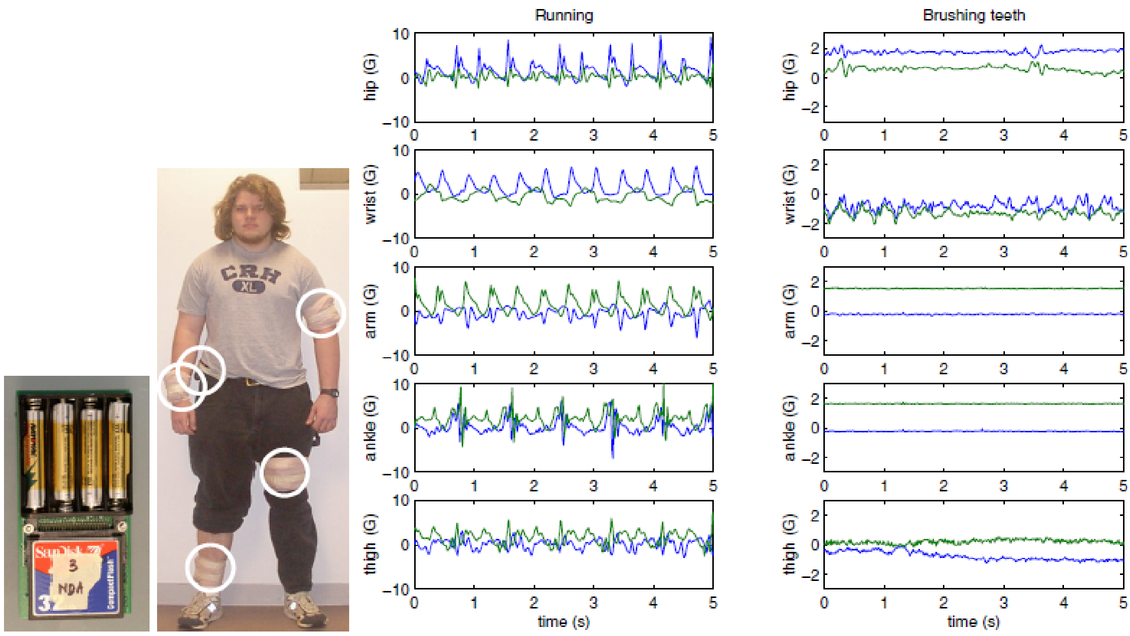
Setting 3 : Data collected for a single subject on one day used as training data, and data collected for the same subject on another day used as testing data.

Setting 4 : Data collected for a subject for one day used as training data, and data collected on another subject on another day used as testing data.

If the data collector and the experimenter was same, the accuracy showed 90% in average although there are some differences among the classifiers. But if the data collector and experimenter was different, the accuracy decreases dramatically to 70%.

This research is meaningful for research using smartphone that they used only one acceleration sensor to recognize various activities. But there are some limitations that the position of the sensor is fixed and is not recognized in real-time.

Another research uses five 2D acceleration sensors[2]. In this research, they recognized activities as walking, running, stay, sitting, watching TV, cycling, eating and reading, and 5 sensors are fixed in forearm, wrist, pelvis, knee and calf. Data was collected from 20 subjects and showed 84% of accuracy in average. We can see that some of the activities are dependent whether the data collector and the experimenter are same or not. [Figure 1] shows the location of the sensor attached and the acceleration signal pattern of two activities.



(a) (b) (c)

[Figure 1] Experimental environment using 2D acceleration sensor and acceleration signal pattern of different activities

This research is different with using a smartphone that it uses 2D acceleration sensor and use more than one sensor. But it is meaningful that it shows high accuracy independent to data collector and experimenter.

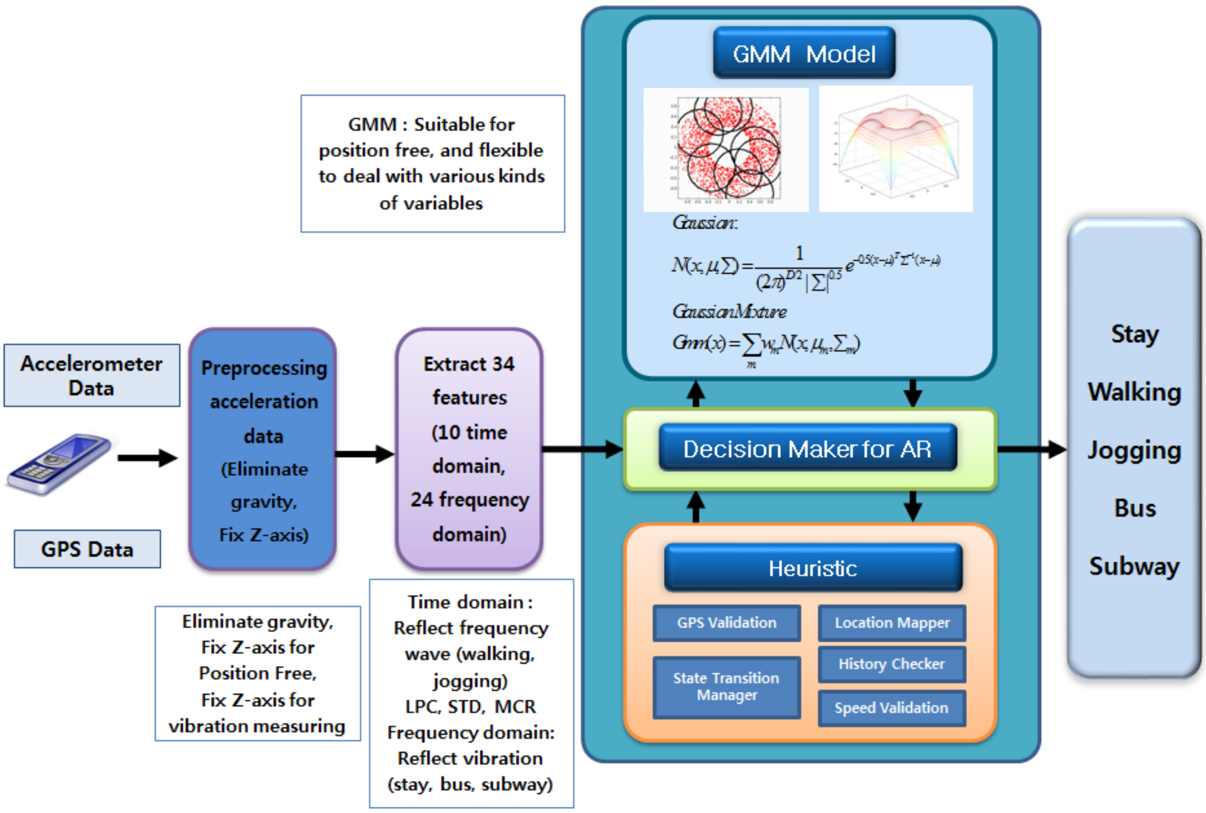
2.2 Smartphone based activity recognition

Shuangquan Wang et al tried to recognize walking, cycling and taking a bus, subway and a car[5]. They used SVM to combine 3 axis of acceleration signals into one to try to recognize activity independent to the orientation of the smartphone, and extracted vertical and horizontal features from this signal. By using vertical and horizontal features, they got 70% of accuracy in average. Bus and subway showed 58.3% and 52.2% respectively, and these two showed the biggest confusion than the others. It seems that this is caused by extracting vertical and horizontal signal from SVM signal. In other words, it seems hard to get an original intact signal if the acceleration signal is combined once.

Another research figured out that using only acceleration data is hard to recognize bus and subway. They used additional sensors in the smartphone with hierarchical structure[10]. They recognized stay, walking, running and shaking using acceleration sensor. Shaking means it is a bus or a subway, and then it uses mic to recognize what it is. They've got high accuracy of 90% in average. But they didn't consider the noise problem when the smartphone is in the pocket. So this method will not guarantee good performance in the real field.

Chapter 3 – Smartphone based activity recognition

In this chapter, we will introduce our methodology to recognize stay, walking, jogging and taking a subway and a bus using a single smartphone without considering position and orientation. [Figure 2] shows the overall view of the research. Detailed contents will be explained below.



[Figure 2] Overall view of the research

3.1 Recognizing walking and jogging using SVM (Signal Vector Magnitude)

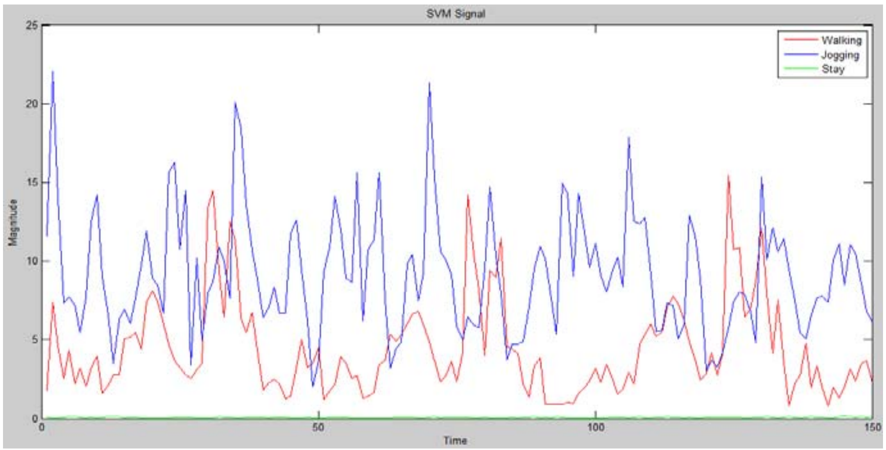
Before the emergence of smartphone, acceleration sensors were attached in the body, and the features are extracted based on the axis that shows the most characteristic pattern according to the activity[2,3]. But this method is useless in smartphone because people don’t carry their smartphone considering orientation. They put it inside the pocket or in the bag or hold it in any direction. So we use SVM to offest the direction information.

SVM combines 3 acceleration signals into one. The formula is shown in <Formula 1>.

<Formula 1> Formula of SVM which combines 3 acceleration signals into one

$$acc = \sqrt{acc_{x_i}^2 + acc_{y_i}^2 + acc_{z_i}^2}$$

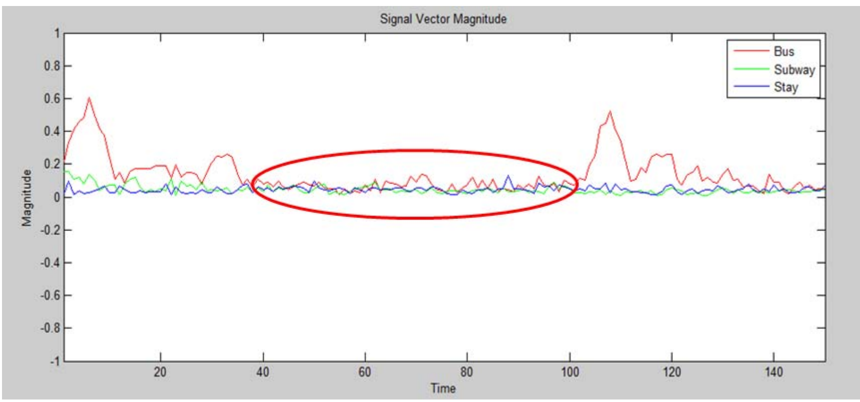
If we extract features from SVM signal, we could easily discriminate stay, walking and jogging which shows big difference of pattern and magnitude as shown in [Figure 3][6]. We also use this method to discriminate stay, walking, jogging and vehicle. Vehicle includes both bus and subway, which it is hard to classify these two by only using SVM.



[Figure 3] Comparison of stay, walking and jogging using SVM

We extract total 10 features from SVM signal such as 5^{th} degree Linear Prediction Coefficient(LPC), 1 LPC error, 3 Standard Deviation(STD) which each is one second, and one Mean Crossing Rate.

But only walking and jogging shows reliable result using SVM while stay and vehicle shows high confusion among them. This is because that stay, bus and subway has no big difference in the signal, and also has no inherent pattern as shown in [Figure 4].



[Figure 4] Comparison of stay, bus and subway using SVM

And also, a small movement will make a big noise to the signal that the magnitude is very weak. To solve this problem, we propose a method in chapter 3.2 by not using SVM but able to discriminate stay, bus and subway irrelevant to smartphone's orientation.

3.2 Activity recognition using vehicle's inherent vibration

In this chapter, we propose a method to discriminate recognizing stay, bus and subway. We first assumed that each vehicle may have their own vibration, and we tried to prove it.

In chapter 3.2.1, we introduce a way to measure vehicle's inherent vibration by preprocessing which we classify the Z-axis from the whole. In chapter 3.2.2, we confirm that we could see the vehicle's inherent vibration in each axis. In chapter 3.2.3, we introduce features that could reflect these inherent vibrations.

3.2.1 Fixing acceleration signal using gyroscope

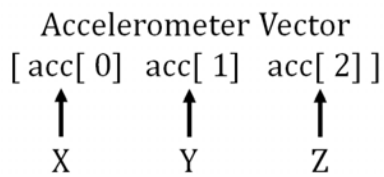
Previous research used SVM to solve the problem when the direction of smartphone is not fixed. SVM solves it by offsetting the effect of axis. But in this case, it could not reflect the own features occurred in each axis. A method was proposed to solve this problem by extracting vertical and horizontal features from SVM signal[5,6]. Even so, it couldn't perfectly restore the original characteristic, and this causes the degradation of the accuracy of vehicle recognition. Therefore, we propose a method to fix the acceleration signal by not using SVM but using gyroscope.

Early released smartphone didn't have gyroscope, but it is basically embedded in current smartphones. Google provides an API for android to catch a change of direction using gyroscope[7]. By using the API, we can get 3 by 3 matrix as shown in [Figure 5.a] from vector information of 3-axis gyroscope.

Rotation Matrix from Rotate Vector

$$\begin{bmatrix} R[0] & R[1] & R[2] \\ R[3] & R[4] & R[5] \\ R[6] & R[7] & R[8] \end{bmatrix}$$

[Figure 5.a] Rotation matrix of transition of each axis of gyroscope



[Figure 5.b] Structure of acceleration data

Acceleration data of each axis's signal vector before fixing are stored in one dimension as shown in [Figure 5.b]. Fixed acceleration signal can be obtained by fixing rotation matrix to original signal. <Formula 2> is used to get fixed X-axis to east and west, and <Formula 3> is used to get fixed Y-axis to north and south, and <Formula 4> is used to get fixed Z-axis to up and down. [Figure 6] shows the concept of fixing acceleration signal.

<Formula 2> Formula to get fixed x-axis signal

$$revAcc_x = \sum_{i=0}^{2} acc_i \times R_i$$

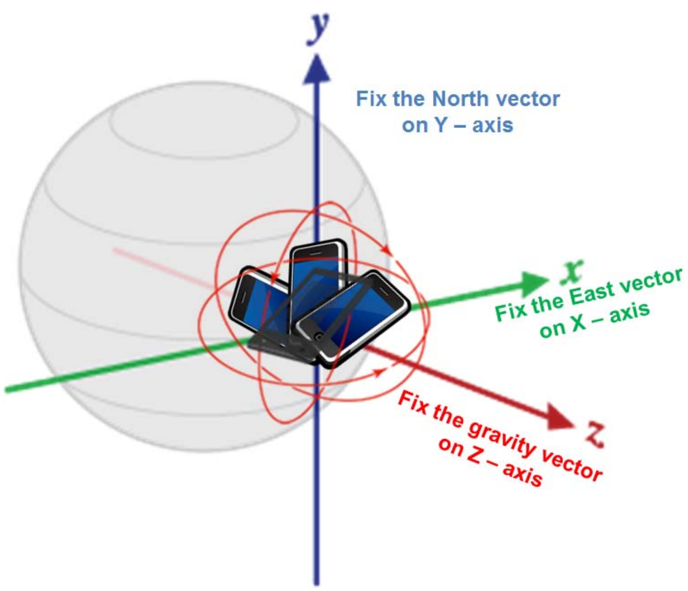
<Formula 3> Formula to get fixed y-axis signal

$$revAcc_y = \sum_{i=0}^{2} acc_i \times R_{(3\times 1)+i}$$

<Formula 4> Formula to get fixed z-axis signal

$$revAcc_z = \sum_{i=0}^{2} acc_i \times R_{(3\times 2)+i}$$

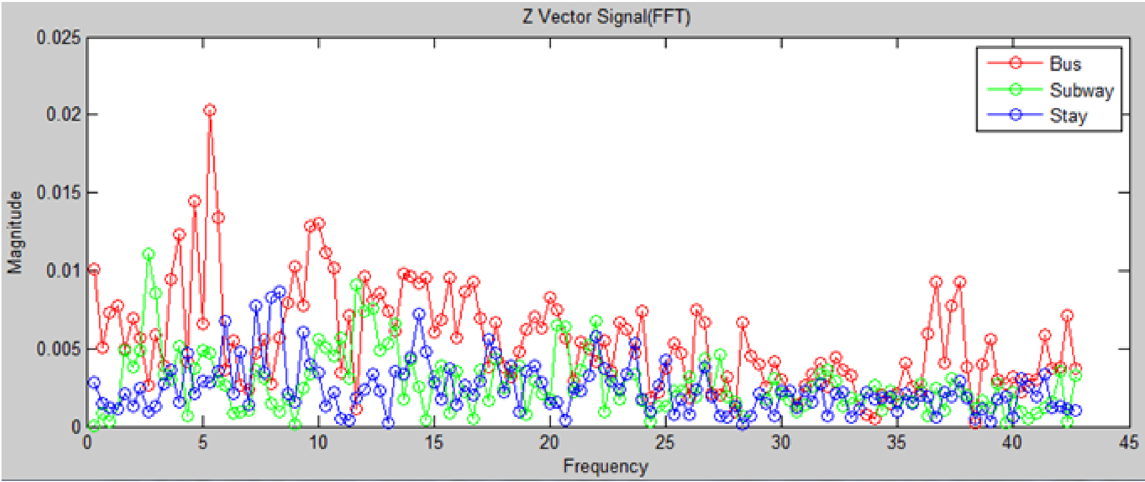
.



[Figure 6] Concept of fixing acceleration signal using gyroscope

3.2.2 Vehicle's inherent vibration occurred from z-axis of acceleration sensor

Vibration occurred in bus and subway comes from up and down axis of acceleration signal, and we may find some features analyzing this. We use Fast Fourier Transform(FFT) to do this. By using FFT, we could change time domain features to frequency domain feature of Z-axis which shows the variation of up and down signal. [Figure 7] shows the stay, bus and subway's signal in frequency domain.



[Figure 7] Signal from Z-axis of stay, bus and subway in frequency domain

The highest frequency happens around 5.3Hz in the bus, and we confirmed the Gaussian distribution between 3 and 7Hz. This might happen because of the variance of the speed of the bus and road conditions. Another feature is that it shows relatively high magnitude than stay and subway.

The highest frequency happens around 2.3Hz in the subway, and we confirmed the Gaussian distribution between 2 and 4Hz. It shows similar magnitude with stay in other frequencies except between 2 to 4Hz.

In stay, there is no particular frequency that shows high magnitude. This is because it doesn't have inherent vibration unlike bus and subway.

3.2.3 Feature extraction to discriminate among stay, bus and subway

As we analyzed in chapter 3.2.2, there are differences among stay, bus and subway in Z-axis. Therefore, we need to extract features reflecting this. For feature extraction on Z-axis, we use total 6 features such as mean, standard deviation, maximum magnitude, frequency of maximum magnitude, ratio of 2~4Hz among the whole frequency and ratio of 3~7Hz among the whole frequency. These features come out from Z-axis.

We also extract same features on X and Y axis. Even though Z-axis reflects the inherent vibration, the magnitude of the signal is very weak and could be easily exposed to noise. So we enhance the accuracy by considering the change of signal such as acceleration of the vehicle. At this moment, we do not need the direction of the vehicle but only the variation of the magnitude. So we use FFT to SVM signal which is only consisted of X and Y signal as shown in <Formula 5>.

<Formula 5> SVM formula combining X and Y axis's signal into one

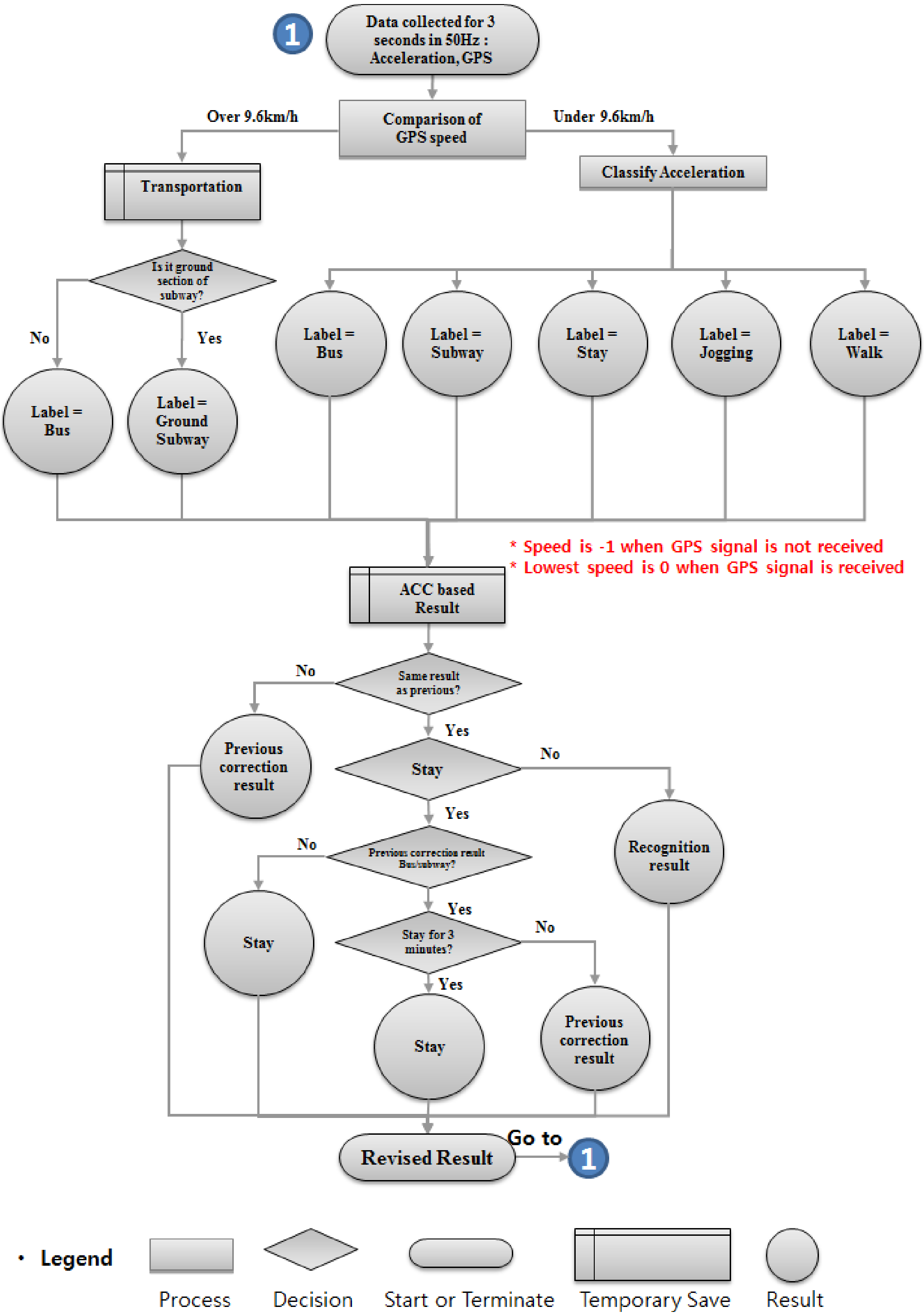
$$acc = \sqrt{acc_{x_i}^2 + acc_{y_i}^2}$$

We also use features as correlation between XY and Z axis in both time domain and frequency domain to see the relation of XY axis and inherent vibration from Z axis.

Finally, we use total 14 features to discriminate stay, bus and subway which are 6 features from each XY and Z axis and 2 correlation features.

3.3 Accuracy correction algorithm for activity recognition

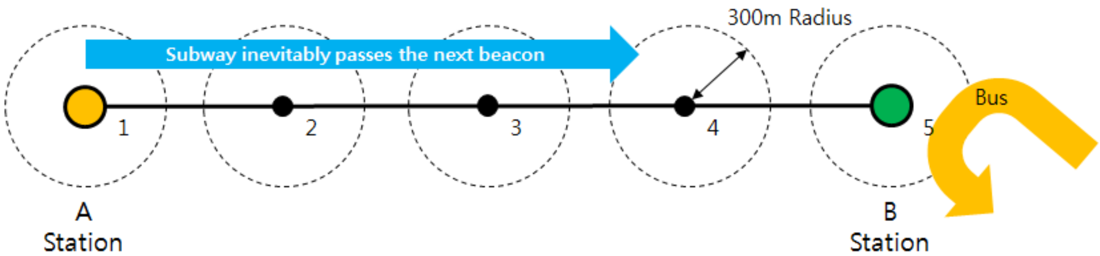
Introducing so far, we basically use acceleration data for activity recognition. But if we use GPS in outdoors, we may discriminate bus and subway more easily. So we get GPS data along with acceleration data with the information of location and speed. Acquired acceleration data are used for feature extraction and activity recognition as described in chapter 3.1 and 3.2, and GPS data is used for correction of recognizing bus and subway. In other words, final activity is chosen by correction using GPS data based on acceleration data result classified by Gaussian Mixture Model(GMM). [Figure 8] shows the overall flow of this process.



[Figure 8] Overall flow of proposed activity recognition method

GPS information varies with surroundings and location. So if the GPS signal is misreceived, both location and speed data will be incorrect. There would be a problem if the speed shows high while the user is in stay. It could misrecognize it as a bus or a subway even if it is stay. To prevent this problem, we should decide a threshold of speed to discriminate stay with bus and subway. We found the threshold and defined the speed as 9.6km/h by comparing the speed whlie putting the smartphone still outside.

Therefore, if the speed from GPS shows more than 9.6km/h, we assume that the current activity is not stay, walk and running, and is either bus or subway. We also consider subway that there are some ground level section it runs. We collected five beacons between two stations, and if the subway passes one beacon to another within 30 seconds, we consider the user is on the subway. If the user doesn't pass one beacon to another within 30 seconds or passes only one beacon, we consider the user is on the bus. Detail explanation is shown in [Figure 9].



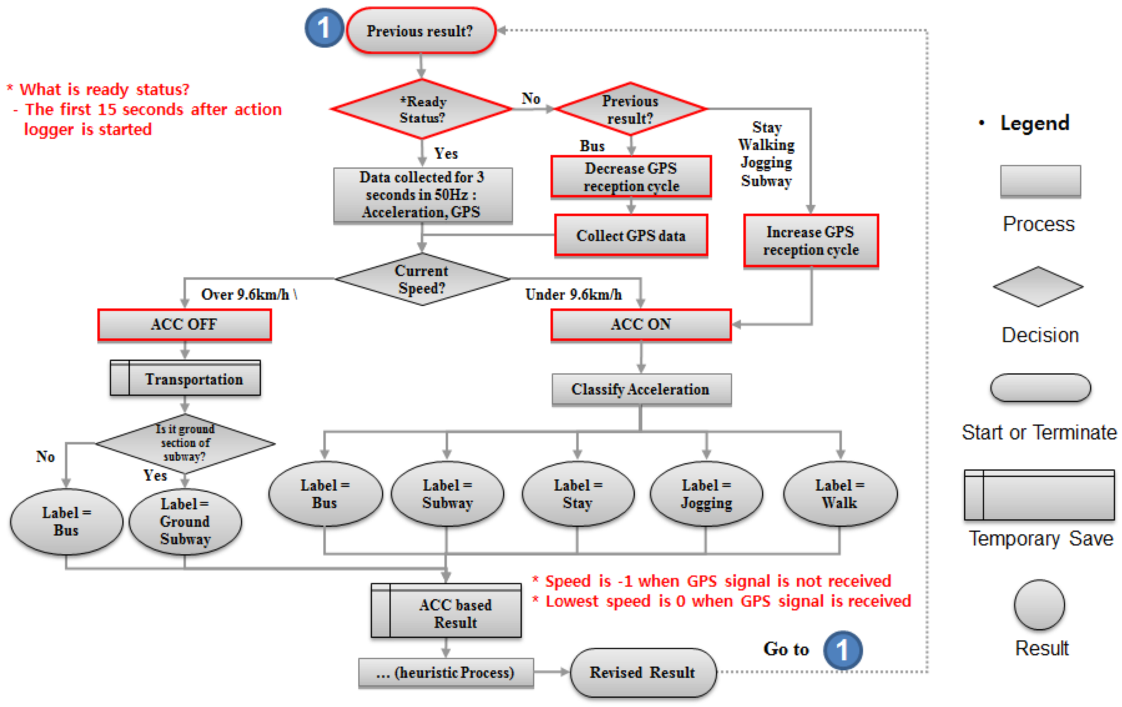
[Figure 9] Discriminating bus and subway using location data from GPS

We also need correction for acceleration data in case the GPS signal isn't received properly. Because we can't guarantee the recognition result from acceleration data is 100% sure. And also the user's don't always put their smartphone calmly. They use it to call, send text or even browse the internet. This remains as noises and would output the result incorrectly. To fix this problem, we decided the final activity to output only if the sequential two results are same. If these two are not same, we output the activity same as the previous output.

If the last recognized result was bus or subway, and the current result is stay, we force the output to be the last recognized vehicle. If we don't use this kind of method, we cannot deal with the misrecognition of stay when it shows low vibration when the vehicle is stopped. If we don't use this correction method, we could easily confirm this phenomenon when testing in the real field. It shows stay frequently when we are still inside the vehicle. When stay is recognized sequentially for 1 minute while the last recognized result was vehicle, the activity transfers to stay from vehicle. This is to prevent misrecognition caused by the vibration occurred by user shaking their leg or handling their phone even if the user didn't take the vehicle.

3.4. Battery saving algorithm

In this research, the recognition module receives GPS and acceleration signal all the time. This consumes unnecessary battery even when one sensor data does not affect the recognition. In average, smartphone consumes 2% of the battery for 2 hours when no sensors are running. But it consumes 5% of the battery when it runs acceleration sensor, and 11% when it runs GPS in the same time. To satisfy the purpose of this research, which means at least we should be able to use this module lasting for a day in real life, we need a solution to save the battery consumption. In this research, we propose an algorithm by dynamically turning on and off or change the cycle of receiving the GPS and acceleration sensor according to activity. If the previous activity was bus and current speed is over 9.6km/h, the module turns off the acceleration sensor, and turns on again when the speed goes under 9.6km/h. And if the activity is recognized as stay, walking, jogging or subway, GPS doesn't affect mainly on the recognition. So we extend the cycle of GPS receive from 1 second to 30 seconds. [Figure 10] shows the battery saving algorithm adjusted to our activity recognition algorithm.



[Figure 10] Proposed battery saving algorithm

Chapter 4 – Experimental result

In this chapter, we introduce our experiment environment and the result of our acceleration sensor and GPS based activity recognition proving our module's superiority. We also define some of the activities which are hard to recognize in the real life, and test them in the real field, which shows good result.

4.1 Experimental environment

We used Samsung's Galaxy S3 (SHW-E210K, SHW-M440S, GT-I9300T) for data collection and testing. One window is consisted of 150 samples which is collected for 3 seconds of X, Y and Z axis in 50Hz respectively, and use the highest speed acquired from GPS data among 3 seconds. Detailed specification of the device used in this research is shown in <Table 2>.

<Table 2> Specification of the device used in the research

Smartphone	Galaxy S3
OS	Android 4.1.2 (Jellybean)
CPU	Quad core 1.2GHz
Memory	1GHz DDR2
sensors	accelerometer, gyroscope, GPS, etc
network	3G, WiFi

4.2 Data collection environment

Smartphone was positioned in the pocket of top and bottoms, in the bag and holding in a hand for data collection. We decided these four positions that these are the frequent positions people carry the smartphone.

We collected stay, walking, jogging and taking a bus and a subway while carrying the smartphone. Especially, we collected both standing and sitting condition while on a bus and subway, and while in stay. We also included the behaviour using the smartphone while in stay such as browsing the internet. This is to prevent misrecognizing different activity while the use manipulates the phone. Total dataset collected is shown in <table 3>.

<Table 3> Total number of collected dataset

Activity	Bus	Subway	Jogging	Stay	Walking	Total
Number of samples	8,942	3,332	1,010	11,404	6,482	31,170

4.3 Evaluation method for performance

Performance evaluation is done by comparing the accuracy of existing research and our proposed method. Existing research combines 3 acceleration signals into one using SVM, and our method uses XY axis and Z axis separately.

We used Gaussian Mixture Model as a classifier. This is based on the fact that each activities has different distribution of features varying from activities, people, noise, etc.

As we collected the data positioning smartphone in different places to guarantee the accuracy irrelevant to the orientation, we need at least four gaussian distribution. And we predicted that there would be one or two distribution in the same activity varying from the gender of the user and personal characteristics. So we used 8 mixtures by expecting 8 gaussian distribution will happen.

We trained using EM algorithm for model generation to select the best distribution within the defined number of mixture.

We checked our module's accuracy by suing 10 Fold-Cross Validation, and classified our collected dataset as shown in <Table 4>. We chose the file randomly for training from the classified training and validation part, and chose the model which shows the best accuracy.

<Table 4> Classification of collected dataset for training

Activity	Training (60%)	Validation (20%)	Testing (20%)	Total
Bus	5,365	1,788	1,789	**8,942**
Subway	1,999	666	667	**3,332**
Jogging	606	202	202	**1,010**
Stay	6,842	2,281	2,281	**11,404**
Walking	3,889	1,296	1,297	**6,482**
Total	**18,701**	**6,233**	**6,236**	**31,170**

This experiment was only to see the accuracy of acceleration sensor based recognition, so the correction algorithm is not used. The result of using correction algorithm will be described in chapter 4.5.

4.4 Evaluation result of performance

In this chapter, we will show the recognition result based on acceleration sensor, and also the result of our proposed method.

First experiment was done using existing method by processing the sensor data with SVM. Based on this, we extract 10 features which are 5^{th} degree LPC, LPC error, 3 STD and 1 MCR. <Table 5> shows the confusion matrix of the result.

<Table 5> Confusion Matrix of activity recognition using SVM

Activity	Bus	Subway	Jogging	Stay	Walking	Average Accuracy
Bus	**0.9214**	0.0414	0.0001	0.0331	0.0040	-
Subway	0.0192	**0.8619**	0	0.1167	0.0021	-
Jogging	0	0	**0.9713**	0	0.0287	-
Stay	0.0452	0.1759	0.000	**0.7713**	0.0075	-
Walking	0	0.0004	0	0.0510	**0.9486**	-
Average Accuracy	-		-	-	-	**0.8954**

Our proposed method additionally uses features extracted from XY-axis and Z-axis, which means the total features used are 24. <Table 6> shows the confusion matrix of the result.

<Table 6> Confusion Matrix of activity recognition using proposed method

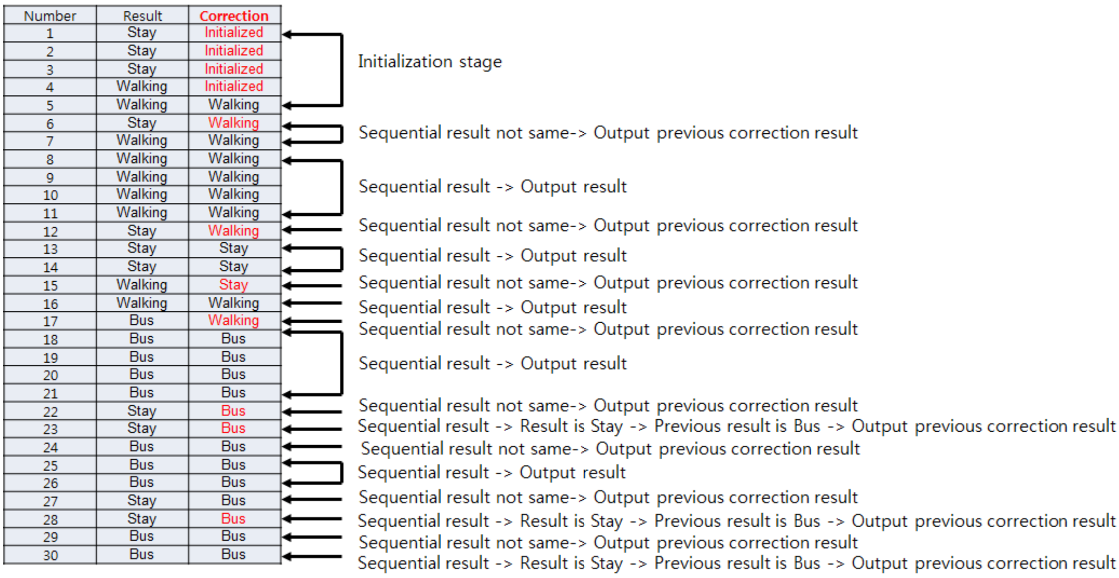
Activity	Bus	Subway	Jogging	Stay	Walking	Average Accuracy
Bus	**0.8912**	0.0119	0	0.0924	0.0045	-
Subway	0.0114	**0.8103**	0	0.1783	0	-
Jogging	0	0	**1**	0	0	-
Stay	0.0100	0.0481	0	**0.9410**	0.0008	-
Walking	0	0.0004	0	0.0510	**0.9486**	-
Average Accuracy	-		-	-	-	**0.9182**

The accuracy of bus and subway showed 92.14% and 86.19% respectively by using SVM which are higher than using our proposed method which the accuracy is 89.12% and 81.03% respectively. But existing method only showed 77.13% accuracy of stay while our method showed 94.10%, which is the dominant activity happens in real life. Using this method is critical to use in the real field. Although our method showed lower accuracy of recognizing bus and subway, the confusion between them showed only 1.2% which is very low. There might have some delay recognizing bus and subway, but it doesn't misrecognize stay as vehicle. The delay of recognizing vehicle is fairly short which may not be a big problem that it takes about 3 minutes moving from one station to another, and we board on them about 15 minutes in average.

4.5 Experimental result using correction algorithm

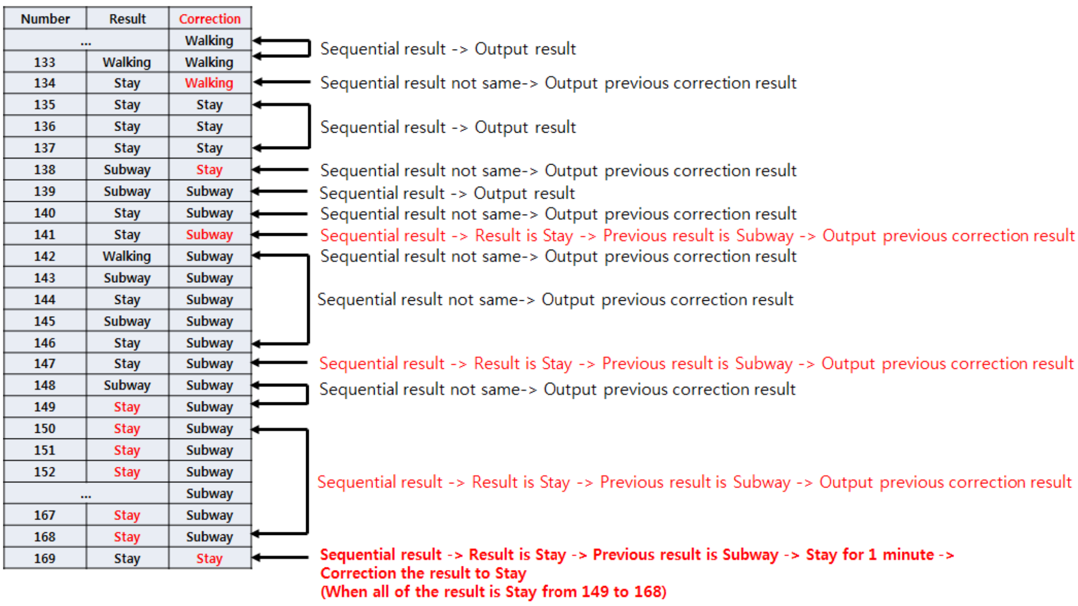
Many kinds of noises will happen living in a real life such as taking out the phone from a pocket or a bag, or the bus passing the speed bump, etc. But looking at the big picture, this takes only a small part. So we didn't consider these problem when collecting the data, but this may cause the misrecognition. To solve this problem, we will use correction algorithm introduced in chapter 3.3.

We will show an example of the experiment done in the real field to show how the correction algorithm works.



[Figure 11] Example of the operation of correction algorithm

[Figure 11] shows the operation of the correction algorithm. Total 30 times of activities were recognized, and the correction was done for each activity. We may see that bus was recognized beginning from 17, but it shows stay in the middle of them. This is occurred by the confusion of stay and bus. But if the correction is used, it shows bus sequentially without showing stay although there is 3 seconds of delay in the beginning.



[Figure 12] Example of correction algorithm when stay is misrecognized as subway

[Figure 12] shows the example of how the correction algorithm works when the activity recognition module misrecognizes different activities to subway. As activity 138 and 139 are recognized as subway sequentially, the engine will finally judge as subway due to correction algorithm. In fact, this is the phenomenon of misrecognition that the real activity is stay. So the recognition result should change from subway to other activity. If walking is recognized sequentially, this error will be revised. But in the example, walking didn't appear sequentially, so the activity didn't changed. Activity from 149 to 169 shows stay sequentially, and if stay comes out sequentially for 1 minute, recognized activity turns to stay and finally escapes the misrecognition.

4.6 Experiment in a real field

To check the accuracy of our proposed method, we conducted the experiment in a real field as shown in <Table 7>. We didn't include walking and jogging since the accuracy of these two activities are high.

<Figure 7> List of test to check the accuracy of recognition in the real field

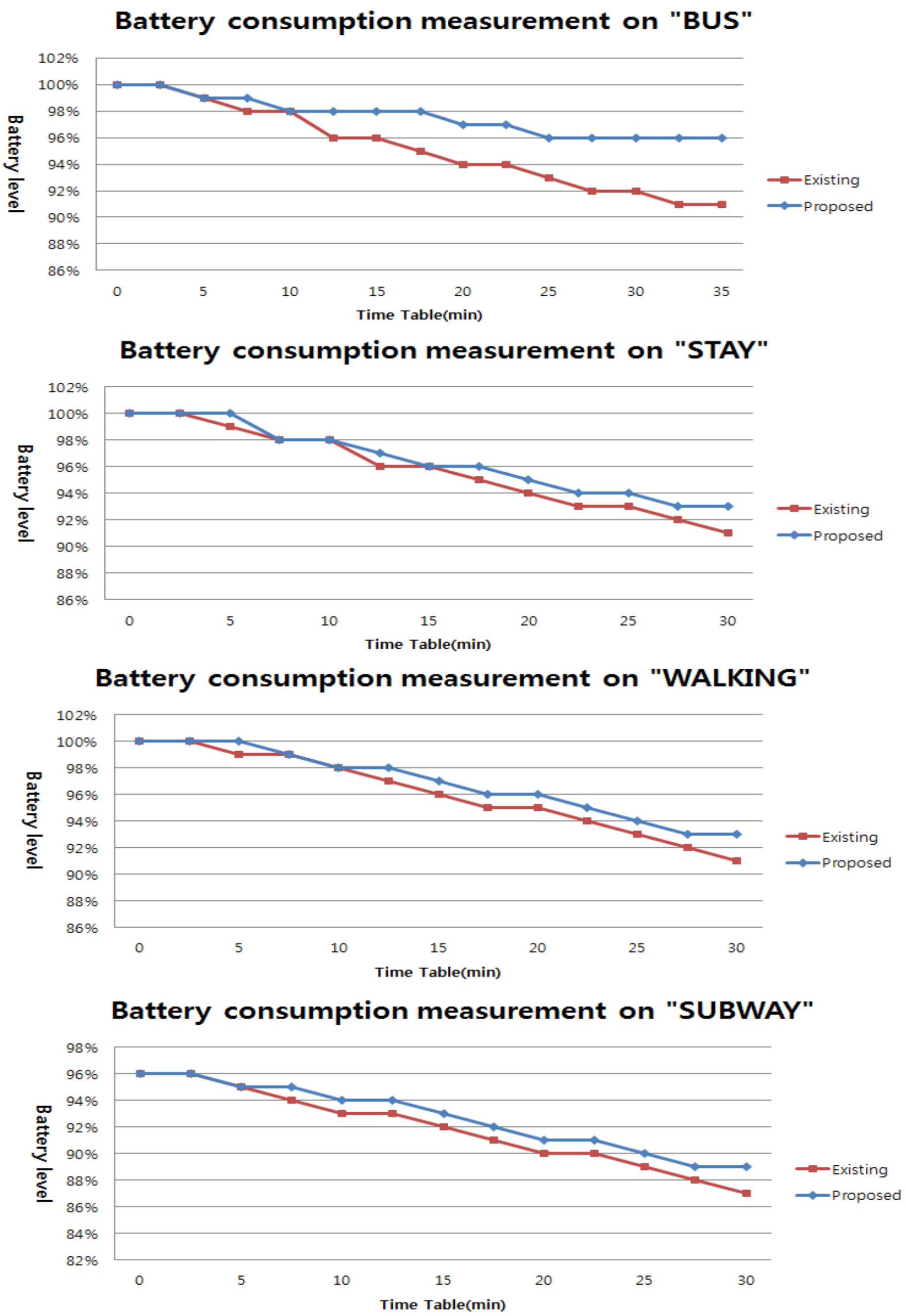
No.	Test List	Result
1	Standing on a subway	33/34
2	Sitting on a subway	13/14
3	Standing on a bus	14/14
4	Sitting on a bus	33/33
5	Standing on an escalator (To see whether it misrecognizes escalator to bus or subway)	45/48
6	Turning on the activity recognition application while inside a subway	46/48
7	Turning on the activity recognition application while inside a bus (To see whether it misrecognizes to different activity while receiving GPS signal)	8/8
8	To see if it turns to different activity while the bus is stopping due to the traffic signal or a bus stop for long time about 2~3 minutes	6/6
9	Riding on a bus for over than 2 hours (To see whether it misrecognizes to subway)	14/14
10	Waiting for a bus or subway at the station for a long time (To see whether it misrecognizes to bus or subway)	52/52
11	Test all activities including transferring to bus and subway for at least 2 hours	19/19

As shown in <Table 7>, our proposed activity recognition technique shows good result in the real field. Most of the test showed exact result except the test 1,2 and 5. These problems are occurred between stay and subway. Unlike bus, subway has small vibration, and this may cause the confusion between bus and subway.

4.7 Experiment of battery consumption

In this chapter, we show the result of our proposed battery saving algorithm. Experiment was done in the real field. And we did not test jogging, considering as walking and jogging are the same activity that these two are done in a same environment.

[Figure 13] on below shows that our battery saving algorithm conserves the energy more than normal way. In bus, battery usage reduced about 5% which is from 9% to 4% for 35 minutes, which means the efficiency increased up to 55%. The reason is that bus spends more time in running than stopping at the bus stop or the traffic, and this makes the acceleration data off. The other activities saved about 2% of battery from 9% to 7% for 30 minutes, which means the efficiency increased up to 22%. As shown in the result, our proposed algorithm can save the battery consumption.



[Figure 13] Comparison of using battery saving algorithm among different activities

Chapter 5 – Conclusion

In this research, we proposed and activity recognition method using acceleration sensor and GPS. We separated 3 axis acceleration sensor data into two to recognize stay/bus/subway with high accuracy which existing works couldn't do. And we proposed a correction algorithm to increase the accuracy in real life by both using GPS data and just before result of activity. We guarantee the accuracy of activity recognition by only using acceleration sensor and GPS from smartphone and also lowered the battery consumption.

Outcome of our research will be greatly helpful to recognize users activity in daily living. We could add additional activities with high accuracy by continuously developing this system.

Chapter 6 – Reference

[1] "Activity recognition" Wikipedia, http://en.wikipedia.org/wiki/Activity_recognition#cite_note-2

[2] Bao, Ling, and Stephen S. Intille. "Activity recognition from user-annotated acceleration data.", Pervasive Computing. Springer Berlin Heidelberg, 2004. 1-17.

[3] Nishkam Ravi et al. "Activity recognition from accelerometer data.", Proceedings of the national conference on artificial intelligence. Vol. 20. No. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[4] "One In Every 5 People In The World Own A Smartphone, One In Every 17 Own A Tablet", Read more: http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10#ixzz2qJT5lj00, 2013.12.15

[5] Shuangquan Wang, Canfeng Chen, Jian Ma, "Accelerometer based transportation mode recognition on mobile phones", APWCS '10 Proceedings of the 2010 Asia-Pacific Conference on Wearable Computing Systems, 44-46p, 2010

[6] Mizell, David. "Using gravity to estimate accelerometer orientation." Proceedings of the Seventh IEEE International Symposium on Wearable Computers (ISWC'03). Vol. 1530. No. 0811/03. 2003.

[7]"SensorManager", Android Developers, http://developer.android.com/reference/android/hardware/SensorManager.html#getRotationMatrixFromVector(float[], float[])