2012 15th International Conference on Network-Based Information Systems

Posture Recognition with G-Sensors on Smart Phones

*Hui-Huang Hsu**, *Kang-Chun Tsai*
Dept of Computer Science and Information Engineering
Tamkang University
New Taipei City, Taiwan
e-mail: *h_hsu@mail.tku.edu.tw,
jknm0510a@hotmail.com

Zixue Cheng, Tongjun Huang
School of Computer Science and Engineering
University of Aizu
Aizu-Wakamatsu, Japan
e-mail: z-cheng@u-aizu.ac.jp, t-huang@u-aizu.ac.jp

***Abstract*—*With the popularity of smart phones in recent years, various sensors on smart phones can be utilized to detect the movement or intention of the smart phone users. In this research, we aim at using the signals collected from the G-sensor in the smart phone to recognize the posture of the user. Signals for sit, stand, walk and run are collected to train an offline neural network as the classifier. After the neural network learns the four postures, we then implement a neural network with the learned connection weights in a smart phone app. The app can record the postures of the user for the whole day and estimate the burned calories accordingly. This app can replace the pedometer to have a more accurate estimate of calorie consumption. Details of the app are presented in this paper. The accuracy of neural networks on posture recognition with G-sensor signals is also verified by five-fold cross-validation.*

Keywords-posture recognition; G-sensors; smart phones; neural networks; ambient intelligence

I. INTRODUCTION

Mobile phones have become a necessity in our daily life. People use mobile phones not only for traditional voice communication, but also for accessing the Internet. Smart phones with higher computing and networking capabilities are now very popular. According to the IDC statistics [1], the smart phone penetration rate surpassed 60 percent in Taiwan in the fourth quarter of 2011. Moreover, various sensors like camera, gyroscope, G-sensor, proximity sensor, light sensor, have become standard equipments on smart phones. Using these sensors to detect the movement or intention of the user for intelligent services is now a major research issue.

Millions of light-weighted applications (apps) have been developed for the smart phone platform. Among them, health management apps are quite popular. In particular, weight control is a major issue in health management since over-weighting is a very serious social problem in developed countries. Eat-less and exercise-more are two major ways to lose weight. It is desired to know self ***activities*** in our daily life. The pedometer can be used to know how much we ***move*** during the day. A lot of pedometer apps can be found in App Store and Google Play. However, we think that a more sophisticated application is possible with the computing capability of the smart phone.

In this paper, we use the signals from the G-sensor in the mobile phone to identify the postures of the user. Four postures are the classification targets – *sit*, *stand*, *walk* and *run*. These four postures quite cover our daily activities. We do not consider exercises like work-out, swimming, and playing basketball at current stage. They can be added at a later time. We build a model based on artificial neural networks (ANN) for such posture classification. A standard feedforward neural network with backpropagation (BP) training algorithm is used [2]. After the posture classification model is built, the neural network parameters are written into the smart phone app. A posture recognition app is thus developed to classify the four states (sit, stand, walk and run) of our daily activities. Calorie consumption of the user can be estimated based on the duration of the four states. We think this kind of app is more informational than simple pedometer apps.

In this research we developed the app for the Android environment. Android is the most popular smart phone OS in the world. According to a report by Gartner research, Android had a market share of 50.9% in the fourth quarter of 2011[3]. This was much greater than Apple iOS with 23.8%. Thus the app is targeted for Android smart phones.

This paper is organized as follows. Section II discusses related work, including posture and gesture recognition via accelerometer and G-sensor. Section III presents the posture recognition app developed in this research. Section IV shows the experimental results of ANN on posture recognition and the implementation of the app. Finally, we draw a brief conclusion and point out future work in Section V.

II. RELATED WORK

There are various applications of accelerometer and G-sensor (a kind of linear accelerometer). Accelerometers are used for gesture recognition in [4] and [5]. They are also used for localization of mobile phones in [6], [7] and [8]. Other applications include end point detection [9] and gait recognition [10].

With accelerometer, mobile phones can even be used to write in air [11], to detect potholes [12] and to detect falls and assess mobility [13]. In [11], a system named PhonePoint Pen is developed to recognize human writing by mobile phone. In [12], the authors use smart phones to detect irregularity on the road. A few data processing algorithms are used in the paper. In [13], the authors use the accelerometer vector norm and a single-threshold algorithm for fall detection. They also discuss the possibility of using smart phones for the Timed-Up-and-Go test.

978-0-7695-4779-4/12 $26.00 © 2012 IEEE
DOI 10.1109/NBiS.2012.135

Wang developed a remote posture monitoring system using two-axis accelerometer in 2004 [14]. Wavelet was used to decompose the accelerometer signals and recognize them into one of the five postures – walk upstairs, walk downstairs, walk, stand and sit. On the other hand, Shih implemented a G-sensor-based pedometer in 2010 [15]. The system finds the maxima and minima of the G-sensor signals and uses thresholds to determine the ups and downs of the signals. However, both of the systems were not on smart phones.

III. Posture Recognition App

In this section, we will introduce the Andoid app developed for posture recognition and calorie consumption estimation. We will first discuss posture recognition by neural networks. We then present the app developed on Android.

A. Neural Network Modeling for Posture Recognition

Fig. 1 shows the neural network model used for learning the postures from G-sensor signals. It is a standard two-layer feedforward neural network. The inputs are the windowed G-sensor signals of five seconds with a sampling period of 0.2 seconds. Hence there are totally 25 processing elements (PEs) at the input layer. The output layer has four PEs which correspond to the four posture states – sit, stand, walk and run, respectively. We add 10 PEs at the hidden layer. This gives the neural network sufficient connection weights to learn the postures.

We first collect G-sensor signals from different subjects. In Fig. 2, we give a set of example signals for the four targeted postures. They can be visually distinguished. The signals are smoothed with moving average and they are normalized before they are sent to the neural network. The neural network model in Fig. 1 is trained off-line with the preprocessed G-sensor signals. After an optimal set of connection weights are obtained, the weights can be used in the Android app for real-time posture recognition.

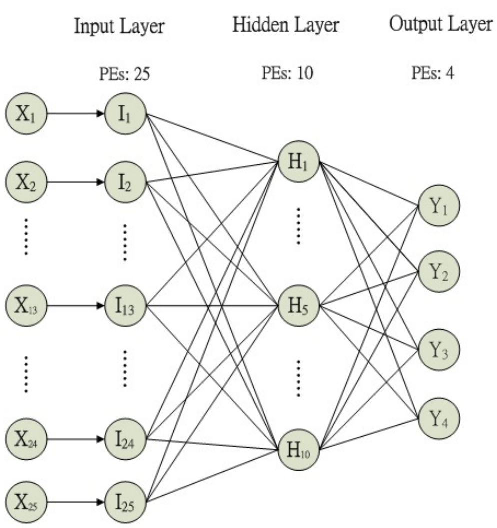

Figure 1. Feedforward neural network for posture recognition

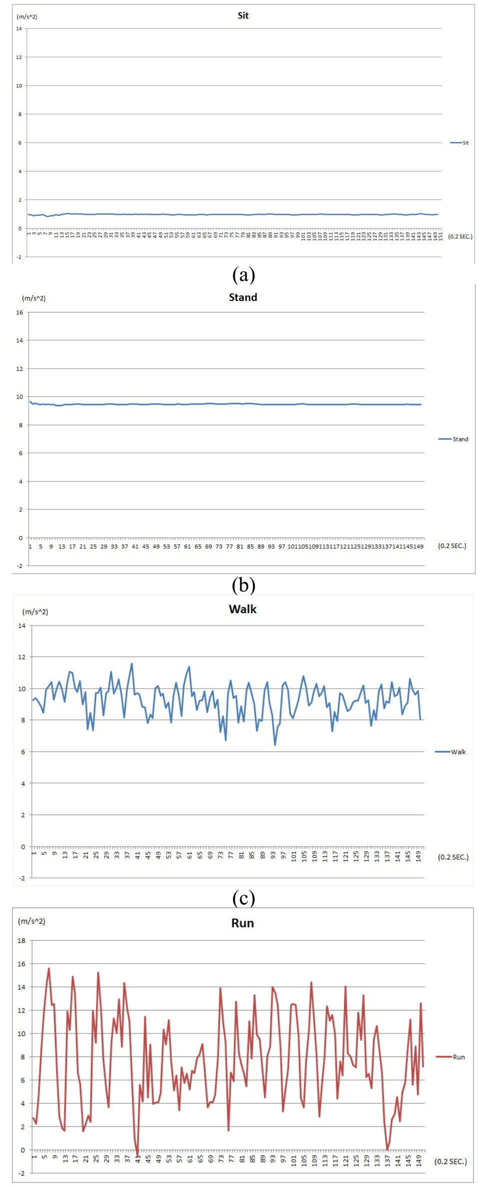

Figure 2. A set of exampleposture signals: (a) sit, (b) stand, (c) walk and (d) run

B. System Architecture

The developed app has a system architecture shown in Fig. 3. It has mainly two modules: *posture recognition* and *history record*. A SQLite database is used to store the posture data. When the user starts the app, it collects the accelerometer (G-sensor) data. The posture recognition module then classifies the data into one of the four postures and save the result to the database. This can proceed for the whole day to record the daily activities of the user till he or she stops the app. In the history record module, the user can query the posture data for a particular date. Calorie consumption is computed and shown to the user along with the posture recognition results.

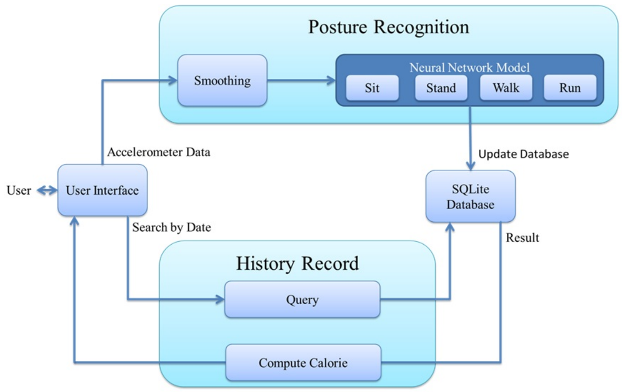

Figure 3. System architecture of the posture recognition app

Calorie consumption is estimated using (1). Calorie consumption for different posture states are of course different. It is basically the weight (in Kg) of the user times the duration of the posture state (in hour) and a posture factor. Here the posture factor is 1.5 for *sit*, 2.5 for *stand*, 5.6 for *walk* and 15 for *run*.

$$calorie = weight \times (1.5 \times sit + 2.5 \times sta + 5.6 \times walk + 15 \times run) / 60 \quad (1)$$

where *sit*, *sta* (*stand*), *walk*, and *run* are the durations of the corresponding posture states in minute.

IV. EXPERIMENTAL RESULTS AND IMPLEMENTAITON

In this section, we will present the experimental results of the neural network in posture recognition. We will show that the neural network is very useful in this classification task. The trained connection weights are then implemented in the app for daily activity recording. The app user can know his/her daily activities and estimated burned calories.

A. Posture Recognition Experiments

We would like to verify the effectiveness of using ANNs for posture recognition. We first collect posture data from six subjects (Table I). All of them are male graduate students in their 20's. The sampling rate is 5 times per seconds. There are totally 20445 data points in the posture dataset, which are about evenly distributed in the four posture states.

TABLE I. POSTURE DATA SET FROM G-SENSOR

Subject \ Posture	Sit	Stand	Walk	Run	Total
A	525	927	383	563	2398
B	1769	1013	1177	854	4813
C	916	1012	720	769	3417
D	600	719	433	557	2309
E	1480	1070	1033	816	4399
F	973	841	623	672	3109
Total	6263	5582	4369	4231	20445

TABLE II. CONFUSION MATRIX FOR POSTURE CLASSIFICATION

Actual \ Assigned	Sit	Stand	Walk	Run	Accuracy
Sit	6304	22	6	9	99.42%
Stand	2	5542	93	9	98.16%
Walk	11	121	4055	88	94.85%
Run	16	23	210	3934	94.05%

The posture data in Table I are used to train the neural network in Fig. 1. Five-fold cross-validation is used to validate the experimental results. The dataset is divided into five portions. For each experiment, one portion of the dataset is left as the test set and the others are used as the training set. The experiments repeats five times with different portions as the test set each time. The experimental results of the five experiments are combined and presented in Table II. The overall classification accuracy is 97 percent (19835/20445). We can see that it is easier to recognize the posture states of *sit* and *stand* and a bit harder to distinguish between the states of *walk* and *run*. Some of the misclassifications occur during the transitions between different states.

We can see that the neural network can learn the posture data from the experimental results in Table II. The neural network is trained again with the whole dataset as the training data. The resulted connection weights are then used to construct the posture recognition module in the app. Since computation for feedforwarding the signals through the neural network is not high, posture recognition can be done in real time on smart phones.

B. App Implementation

An Android app is implemented in this research. It fits the Android 2.1 platform and is downward compatible to Android 1.6. The app was developed under Eclipse IDE for Java EE Developers and tested on HTC Desire smart phone.

The user interface is shown in Fig. 4. The upper part of the user interface is simply a *start* button. When the user touches the button, the app begins to collect G-sensor signals and classify the signals into the posture states in real time. Durations of the posture states are accumulated in the SQLite database. The storage space is minimal since only four numbers are recorded. Moreover, the *start* button turns into a *stop* button after it is touched. The user can stop the process by touching the *stop* button. Durations of the posture states are recorded with a date label which is used for subsequent searches.

When the user wants to know the estimate of his/her calorie consumption of a certain date, he or she can simply input his/her weight (in Kg) and the desired date, and touch the *show* button. The accumulated posture durations (in minute) are then displayed at the bottom along with the estimated total calorie. The total burned calorie is calculated using (1). A pop-up menu is also designed for the date input (Fig. 5).

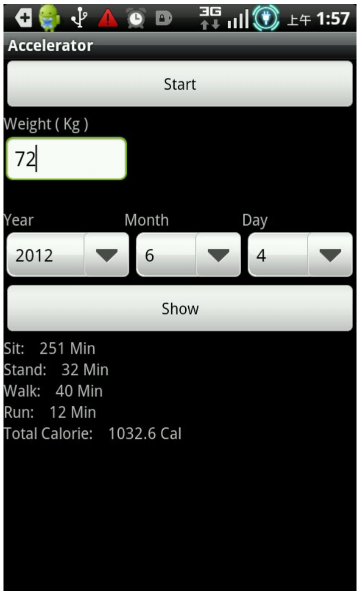

Figure 4. User interface on smart phone

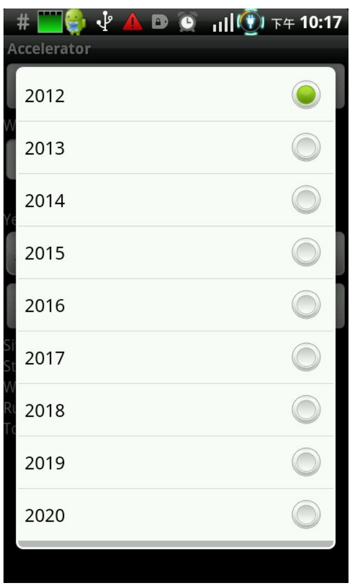

Figure 5. Pop-up menu for date input

V. CONCLUSION AND FUTURE WORK

A posture recognition app is developed and presented in this paper. It can be used to estimate the burned calorie from the daily activities of the smart phone user. It is a very convenient tool since a lot of people carry smart phones nowadays. No extra equipments are needed. The user can be aware of his/her daily activities in a better way and possibly move more to enjoy a healthier life.

Posture recognition with G-sensor signals using neural networks results in a high classification accuracy. In the future, we think there are two possible ways to further improve the posture recognition task. First, we used G-sensor signals from only six male subjects of about the same age. More signals from different sex and ages should be used to train a neural network model for the general public. Secondly, a *personalized* modeling mechanism can be developed. The user's activity signals are collected and used to train a personalized neural network model for posture classification. This should be able to make the classification accuracy nearly perfect.

REFERENCES

[1] International Data Corporation, http://www.idc.com/home.jsp (last accessed Jun 06, 2012).

[2] L.M. Fu, Neural Networks in Computer Intelligence, McGraw-Hill, 1994.

[3] "Worldwide Smartphone Sales to End Users by Operating System in 4Q11," http://www.gartner.com (last accessed Jun 07, 2012).

[4] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo and S. D. Marca, "Accelerometer-Based Gesture Control For A Design Environment," *Personal and Ubiquitous Computing, vol. 10, no. 5*, 2006, pp. 285–299

[5] A. Akl and S. Valaee, "Accelerometer-Based Gesture Recognition via Dynamic–Time Warping, Affinity Propagation, & Compressive Sensing," *Proc. the 35th* IEEE *Int. Con. on Acoustics Speech and Signal Processing*, 2010, pp. 2270–2273.

[6] A. Ofstad, E. Nicholas, R. Szcodronski, and R. R. Choudhury, "AAMPL: Accelerometer Augmented Mobile Phone Localization," *Proc. the 14th* Annual *Int. Conf. on Mobile Computing and Networking*, 2008, pp. 13–18.

[7] I. Constandache, R. R. Choudhury and I. Rhee, "Towards Mobile Phone Localization without War-Driving," *Proc. the 9th Int. Conf. on Ubiquitous Computing,*, 2010, pp. 1–9.

[8] M.Youssef, M.A. Yosef and M.E.Derini, "GAC: Energy-Efficient Hybrid GPS-Accelerometer-Compass GSM Localization," *Proc. Global Telecommunications Conference*, 2010, pp. 1–5.

[9] J. G. Lim, S. Y. Kim and D. S. Kwon, "Pattern Recognition-Based Real-Time End Point Detection Specialized For Accelerometer Signal," *Proc.* IEEE/ASME *Int. Conf. Adv. Int. Mechatron.*, 2009, pp. 203–208.

[10] Y. Liu, Y. E. Li and J. Hou, "Gait recognition based on MEMS accelerometer," *Proc. the 10th Int. Conf. on Signal Processing*, 2010, pp. 1679-1681.

[11] S. Agrawal, I. Constandache, S. Gaonkar, R.R. Choudhury, K. Caves and F. DeRuyter, "Using Mobile Phones to Write in Air," *Proc. the 9th Int. Conf. on Mobile Systems, Applications, and Services*, 2009, pp. 15–28.

[12] A. Mednis , G. Strazdins, R. Zviedris, G. Kanonirs and L. Selavo, "Real Time Pothole Detection using Android Smartphones with Accelerometers," *Proc. the 12th Int. Conf. on Distributed Computing in Sensor Systems and Workshops*, 2011, pp. 1–6.

[13] C. Tacconi, S. Mellone and L. Chiari, "Smartphone-based Applications for Investigating Falls and Mobility," *Proc. the 5th Int. Conf. on Pervasive Computing Technologies for Healthcare,* 2011, pp. 258–261.

[14] Y.C. Wang, Remote Posture Monitoring Using a Two-Axis Accelerometer, Master thesis, National Chiao Tung University, Taiwan, 2004.

[15] C.C. Shih, The Implementation of a G-Sensor-based Pedometer, Master thesis, National Central University, Taiwan, 2010.