The User Identification System Using Walking Pattern over the ubiFloor[1]

Jae-Seok Yun*, Seung-Hun Lee**, Woon-Tack Woo**, Je-Ha Ryu*

*Department of Mechatronics, KJIST, Gwangju, Korea
(Tel : +82-62-970-2417; E-mail: {zapper, ryu}@kjist.ac.kr)
**Department of Information & communications, KJIST, Gwangju, Korea
(Tel : +82-62-970-3157; E-mail: {slee, wwoo}@kjist.ac.kr)

Abstract: In general, conventional user identification systems require users to carry a TAG or badge or to remember ID and password. Though biometric identification systems may relieve these problems, they are susceptible to environmental noise to some degree. We propose a natural user identification system, ubiFloor, exploiting user's walking pattern to identify the user. The system identifies a user, while tracking the user's location, with a set of simple ON/OFF switch sensors or equipments. Experimental results show that the proposed system can recognize the registered users at the rate of 92%. Future improvement in recognition rate may be achieved by combining other sensors such as camera, microphone, etc.

Keywords: ubiFloor, stepping features, walking patterns, user identification

1. INTRODUCTION

In general, when using a computer, a user is required to give an ID and a password to be authorized so far, but the need of various identification systems is increased with the deployment of ubiquitous computing. RF TAG systems like Active Badge[1], BAT[2], and RADAR[3] that compare a unique ID within each sensor to identify the owner provide robust results regardless of noise around a user because it sends data through radio frequency. However the user must carry a badge or tag to be identified. In addition, issuing sensors for temporary visitors is cumbersome.

To complement the defects of the user identification system based on carrying or wearing sensors, the biometric identification has been developed. It analyzes the pattern of an iris, face, voice, and fingerprints, and therefore users are not required to carry any sensors. However, biometric identification systems are not yet perfect[4]. In case of camera-based systems, for example, occlusions, shadow, and the intensity of light affect recognition result.

New identification systems such as Smart Floor[5] and Active Floor[6] exploiting the user's walking pattern emerged to make up for the weak point of biometric systems. They identify users by the weight and stepping features such as the pressure of toe and heel using several load cells arranged on the floor tile. They do not need much computational resources due to the simple algorithms for identification. However, users may not feel this system is natural because participants are asked to step on the particular area of a tile. Also, high cost of load cell makes the systems expensive.

Our proposed system adopts the ubiFloor shown in Figure 1 and identifies users by the walking pattern. From the ubiFloor consisting of simple ON/OFF switch sensors, the user's step positions are successively transmitted to a host PC in real-time. We have developed two software programs to identify the user based on received data sets. First is the feature extracting software from received data sets, second is the user identification software with deploying extracted features.

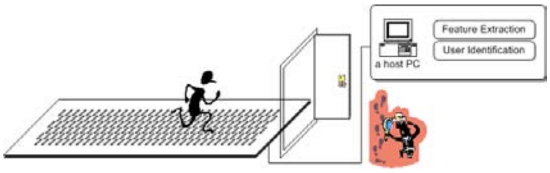


Fig. 1 The overview of user identification with the ubiFloor

The proposed system has the following features. It uses the position of several steps as user's walking features instead of the pressure of one step which was proposed in [5][6]. Since users walks on the ubiFloor as they usually do, it provides a natural and comfortable interface to the users. In addition, the algorithm is not complicated nor computationally intensive. Furthermore, the ubifloor can be used as an interface to control a computer for various purposes. For example, if the system can give visual or audio feedback to a user, the user can interact with a computer by moving steps[7]. Due to the simplicity of the proposed system, it can be reconfigured in various shapes and can be extended easily.

This paper is organized as follows. In Section 2, the ubiFloor system is explained. User identification is described in Section 3. Experimental results are shown in Section 4 and we conclude and suggest the future works in Section 5.

2. THE UBIFLOOR SYSTEM

2.1 System configuration

In the proposed system, we have developed the ubiFloor system to reliably gather walking patterns of user. The ubiFloor system consists of 144 low-cost ON/OFF switch sensors on a mat and a DAQ board that acquires and transmits state of each sensor to a host PC. A switch sensor measures 14cm by 2.5cm. Figure 2 shows the shapes of a sensor and the corresponding electric circuit.

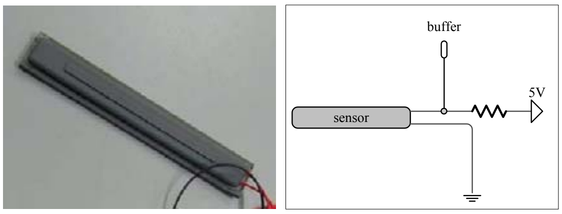


Fig. 2 An ON/OFF switch sensor and electric circuit

As shown in Figure 3, we have placed four sensors on a cell measuring 30cm by 30cm in order to cover one user's footstep

[1] This work was supported by University Research Program (2001-2003) of MIC in Korea.

considering the standard foot size.

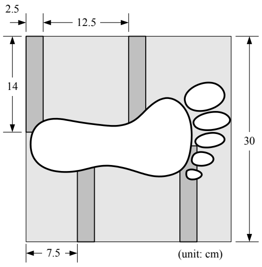


Fig. 3 One cell of the ubiFloor

Figure 4 shows implemented ubiFloor. We have grouped all of 144 sensors in 12 by 3 cells. As a result, the ubiFloor provides a walking area of 3m by 1m as shown in Figure 5.

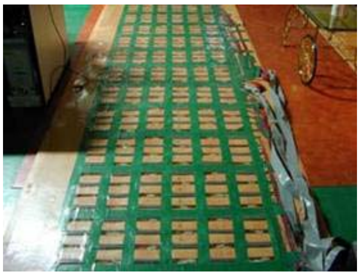


Fig. 4 The implemented ubiFloor

The overall structure of the ubiFloor system is shown below in Figure 5. All of the 144 switch sensors are respectively connected to eighteen 8-bit buffers in groups of eight, i.e. two adjacent cells, to inform the 80KC196 microprocessor of each own state through 50-pin cables. We have deployed two 16-channel DEMUXes to properly intermediate between eighteen 8-bit buffers and an 8-bit data port of the 80KC196 microprocessor. Lastly, we have connected the microprocessor with the COM1 port of a Host PC thorough a RS232 serial cable.

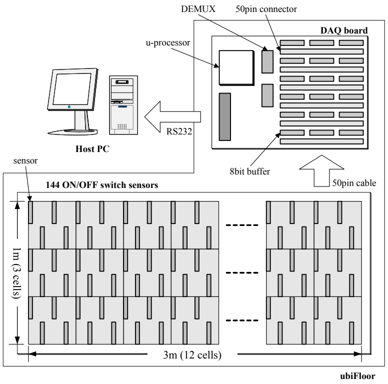


Fig 5 The structure of the ubiFloor system

2.2 Data acquisition and transmission

To acquire one data set of the ubiFloor, the microprocessor reads its 8-bit data port 18 times while sending control signal to 2 DEMUXes and selecting the corresponding buffer one by one. After gathering one data set from the ubiFloor, the microprocessor compares the current data set with previous one. If there is no change between both data sets, the microprocessor decides to gather next new data set. In the opposite case, the microprocessor sends the changing information to the Host PC.

For the efficient data transmission, we have made the concept of a block in groups of eight sensors, i.e. two adjacent cells. Therefore, as explained in the previous section, one block of the ubiFloor is corresponding to one buffer of the DAQ board. Figure 6 shows all indexed blocks of the ubiFloor and eight numbered sensors of one block.

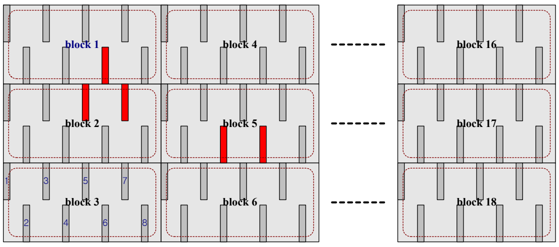


Fig. 6 Block and sensor indexing

Structure of transmitting data format is shown in Figure 7. The BLOCK # is used to express the specific index of the block to include currently pressed sensors, while the DATA is used to represent that which sensors in the specific block are pressed in the form of a hexa-decimal value. The SYNC_BIT is used to inform the Host PC of finishing sending one data set. For example, if sensors are pressed as shown Figure 6, resulting transmitting data is expressed as shown below in Figure 7.

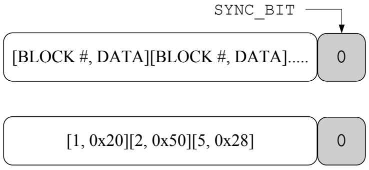

SYNC_BIT
[BLOCK #, DATA][BLOCK #, DATA].....
0
[1, 0x20][2, 0x50][5, 0x28]
0

Fig. 7 The structure of transmitting data format

When the Host PC receives a SYNC_BIT, it crops the SYNC_BIT and converts cropped data set into the string indicating the block number and the number of the pressed sensors. Lastly, the Host PC inserts the corresponding time on receiving the SYNC_BIT in the head of the data string. For example, if the microprocessor sent the data as shown in Figure 7, the Host PC write the string such as "`TIME` [1 6] [2 5] [2 7] [5 4] [5 6]" to disk. Figure 8 shows an example on receiving several data sets.

```
.
.
0:6221259     [11 2]
0:6684898     [11 2]   [13 3]
0:6993333     [11 2]   [13 3]   [14 3]
0:7121490     [13 3]   [14 3]
0:7372329     [14 3]
.
.
```

Fig. 8 An example of received data sets

3. USER IDENTIFICATION WITH UBIFLOOR

3.1 Walking pattern extraction

The software we have developed for extracting walking pattern from data sets falls into two categories: step-feature extracting and walking-feature extracting.

The step-feature extracting software is used to search out all footsteps in all of received data sets during walking once over the ubiFloor, and then, calculate the step-features of each footstep based on the data sets. Figure 9 shows an example of searching out 6 footsteps in data sets.

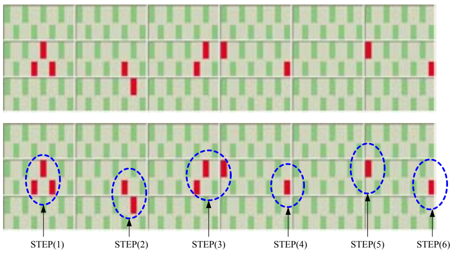


Fig. 9 An example of extracting footsteps

In modeling each user's footsteps, we have chosen four step-features to use as inputs for next walking-feature extracting procedure. For our step-features we are using: the X and Y indices of the backmost sensor in a footstep, the number of pressed sensor in a footstep, and the class of footstep. Figure 10 shows all of possible classes of footstep.

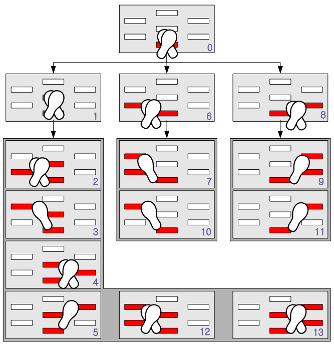


Fig. 10 The step classification

After finishing the step-feature extraction, the resulting features are supplied as the inputs to the walking-feature extraction procedure. For our walking-features we are using:

- the FX and FY: physical coordinates,
- the com_FX and com_FY: compensated FX and FY coordinates based on each class of step,
- and the number of pressed sensor in each step.

Figure 11 shows the FX, FY, compensated FX, and compensated FY features.

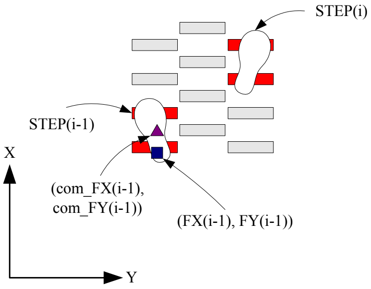


Fig. 11 Adopted walking features

3.2 User identification

For user identification, we have adopted a Multilayer Perceptron with a Neural Network. If it is trained iteratively, the neural network exhibits some capability for generalization beyond the training data to produce approximately correct results for new cases that were not used for training. Neural Network is useful for solving highly non-linear problems and it takes less time and cost than other algorithms. The learning algorithm used in this system is the popular backpropagation of which the direction of updating weights is the opposite direction of processing data.

The Neural Network architecture in Figure 12 consists of three layers; N_1 input layer with P_1 neurons, N_2 hidden layer with P_2 neurons, and N_3 output layer with M neurons. The N_1 input neurons are a feature space representation of steps. If the number of step features is F_N and the number of steps is S_N, then the total number of input neurons, P_1 is equal to F_N x S_N. The number of N_2 hidden layer P_2 neurons is selected by experiments and the number of N_3 output layer P_3 neurons is the number of users to be identified.

For learning algorithm, backpropagation is employed. Since we are doing supervised classification, it is assumed that the class of each input training pattern is known. The output nodes employ a pure-linear transfer function which performs a linear mapping, while hidden nodes adopt a tangent-sigmoid transfer function whose outputs converge into between -1 and 1. Figure 13 shows the layer diagram of the Neural Network in where *IW* is Input Weight, *LW* is Layer Weight, and b denotes bias.

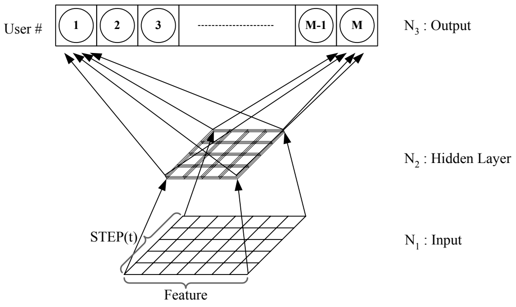
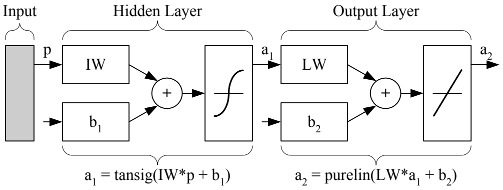


Fig. 12 Structure of Neural Network for user identification



Input
Hidden Layer
Output Layer
p
IW
b_1
a_1
LW
b_2
a_2
a_1 = tansig(IW*p + b_1)
a_2 = purelin(LW*a_1 + b_2)

Fig. 13 Layer diagram

The procedure of training and test is as follows: Firstly, the features input are preprocessed to downscale between -1 and 1. These downscaled features are the input to the hidden layer for processing. As a result of simulation, the system takes the maximum value from its output as:

$$\text{User\_Number} = \max(O_1, O_2, \ldots, O_M) \qquad (1)$$

where O_i is the value of i-th neuron in the output layer.

4. EXPERIMENTS

4.1 Experimental population and condition

We gather walking samples from 10 subjects, 8 male and 2 female. The ages of subjects are from 25 to 31 and heights are from 165 to 179cm. Each participant walked 50 times across the ubiFloor. In total, we gathered 500 walking samples. Depending on the user's stride length, it took 5-6 steps to cover the floor. Out of these steps, we have used data of first five steps only.

We used the MATLAB Neural Network Toolbox. In our experiment, we have deployed Scaled Conjugate Gradient and Conjugate Gradient with Powell/Beale Restarts and compared results because they showed higher recognition rate than other learning functions. In walking samples, we divided each user's sample into 40%, 20%, and 40% for training, validation, and test in random indexing. For efficient learning in training mode, we did not train the whole training samples by testing the trained neural network with validation samples after each training. If it reaches the goal, it stops learning automatically.

4.2 Walking feature and recognition accuracy

To verify the importance of each feature we have adopted in above section, we have classified feature set for input layer. Table 1 shows the combination of features. In Case 1, all of the 144 sensor information is input into the Neural Network without being processed. Based on the result of Case 1, we compared other results from Case 2 to 5.

Table 1 Classification of feature sets

Case	Feature sets	# of input neuron
1	Raw data	144
2	FX, FY	10
3	FX, FY, and # of pressed sensors	15
4	com_FX, com_FY	10
5	com_FX, com_FY, and # of pressed sensors	15

We have been able to increase recognition rate while decreasing learning time as follows. At first, as the number of neurons in hidden layer increases, it increases the performance but needs more computing time. We have changed the number of neurons in hidden layer while other parameters are set fixed. Figure 14 is the performance curve for the number of hidden node. It shows that the recognition rate is saturated roughly to 80% at 48 neurons in hidden layer. Another parameter which controls the early stopping is epochs. Figure 15 shows the convergence of Mean Square Error(MSE) as the value of epochs increases. Roughly after 800 epochs, the MSE is converged to 10^{-2} and this value is set to the value of 'goal' which is used as the other control parameter to stop training.

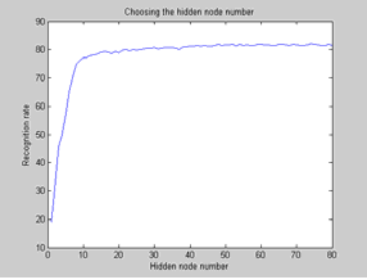


Fig. 14 Performance curve for the number of hidden node

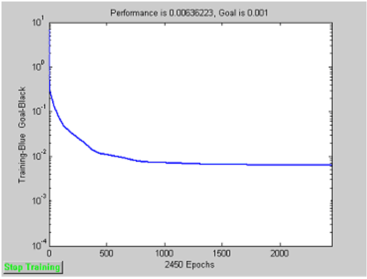


Fig. 15 MSE curve for iteration number

Figure 16 shows the simulation result of 10 user samples. The condition of this Neural Network is as follows: The deployed features are com_FX and com_FY. The number of hidden nodes is 48, and the seed value for weights and biases is fixed. The upper graph shows the original test samples and the lower one represents simulated result from trained Neural Network. It is the case of best performance (92% - 16 out of 200 samples were incorrectly identified). From this figure we can know the similar walking patterns of different users.

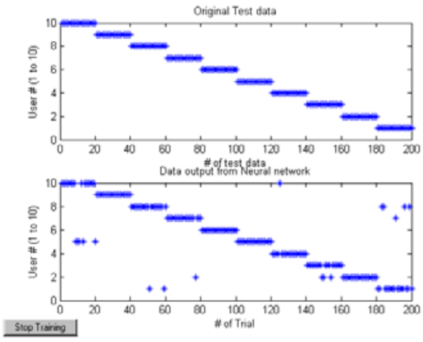


Fig. 16 Simulation graph for 10 users

Table 2 shows the comparison of recognition rate. In this table, the recognition rate is not the maximum result from one simulation but the average of 10 simulations while changing the value of seed which determines the initial value of weights and biases in training function. From our experiment, the compensated coordinated com_FX and comFY results in best recognition rate, 92%. In addition, we have found that by increasing the resolution of the ubiFloor, the performance of recognition could be improved. Even though the number of pressed sensors increased the performance with physical coordinates FX and FY, on the contrary, it decreases the recognition rate together with com_FX and com_FY.

Table 2 Comparison of recognition rate

Case	Feature sets	# of input neuron	Recog. Rate (%)
1	Raw data	144	84.0
2	FX, FY	10	80.5
3	FX, FY, and # of pressed sensors	15	86.0
4	com_FX, com_FY	10	92.8
5	com_FX, com_FY, and # of pressed sensors	15	86.0

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the ubiFloor, a system to identify users only with the walking features of several steps. Our proposed user identification system can provide the natural interface since it does not require users to carry sensors and to pay attention to anything. The performance of the system shows about 92% recognition rate with the properly designed recognition algorithm. This recognition rate may not be enough to be used successfuly in public places. For more robust and increased recognition rate, vision or voice recognition system may be combined with the proposed walking system. The alternative way is to increase the resolution of the ubiFloor. In addition, centralized system architecture may need to be decentralized since all sensors are connected together to a DAQ board which can cause a problem when there is disconnection on wire or the DAQ board. To solve this problem, the design of the ubiFloor can be decomposed into modular units. In order for the ubiFloor to be deployed, each unit needs to have capability of network and processing.

REFERENCES

[1] A. Harter and A. Hopper, "A Distributed Location System, for the Active Office," *IEEE network*, Vol. 8, No. 1, 1994.

[2] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The Anatomy of a Context-Aware Application," *ACM Mobicom*, 1999.

[3] P. Bahl and V. N. Padmanabham, "RADAR: An In-Building RF-Based User Location and Tracking System," *IEEE Infocom*, 2000.

[4] A. Jain, L. Jong, and S. Pankanti, "Biometric identification," *Communication of ACM*, Vol. 43, 1998.

[5] R. Orr and G. Abowd, "The smart floor: A mechanism for natural user identification and tracking," *ACM CHI 2000 Conference proceedings*, pp. 1-6, 2000.

[6] M.D. Addlesee, A.H. Jones, F. Livesey, and F.S. Samaria, "ORL Active Floor," *IEEE Personal Communications*, Vol. 4, No. 5, pp. 35-41, 1997.

[7] S. Lee and W. Woo, "The Interactive Dancing Floor: The Smart Floor Plays Music," *WIAMIS* pp. 328-331, London, U.K, 2003.