Inhabitants Tracking System in a Cluttered Home Environment Via Floor Load Sensors

Wen-Hau Liau, Chao-Lin Wu, *Member, IEEE*, and Li-Chen Fu, *Fellow, IEEE*

***Abstract*—Home automation systems should evolve to the next phase in which they are aware of contexts, because providing services based on contexts will upgrade the service quality, thus making people more comfortable and home living more convenient. In particular, location is a piece of important and useful context information for seeking appropriate services, as well as providing them to people living in a home. So far, there have been several studies focused on tracking inhabitants in smart home environments. However, these approaches are often intrusive or require inhabitants to wear some sort of devices, which may make the inhabitants uncomfortable or even inconvenient. This problem could devastate the ultimate goal, which is to provide convenient services, and hence cause such approaches somewhat controversial. In this study, we utilize a number of load sensors together to construct a sensory floor on which exerted pressure can be detected and cover its surface with wooden flooring as in a normal home environment. Although the wooden flooring provides a flat surface for inhabitants to walk on, it also causes clutter in the sensor readings, which lead to difficulty in clearly identifying the location(s) of pressure source(s). Thus, we apply Probabilistic Data Association technique and LeZi-Update approach to analyze the cluttered pressure readings collected by the load sensors so as to determine the positions of the inhabitants, as well as to track their movements. With our nonintrusive approach, there is no need for inhabitants to wear any devices, and this also solves the cross-walking problem in the home environment.**

***Note to Practitioners*—This system aims for detecting the location of multiple inhabitants in the home environment. We adopt the sensory floor as our tracking sensor, which consists of many blocks, each with a load cell to collect the human body weight. These blocks are covered with conventional wooden flooring to provide a flat surface on which inhabitants can walk freely. After collecting data, we apply mathematical techniques to analyze them, thus determining inhabitants' locations and tracking their movements. The limitation is that the system cannot differentiate different persons if they have close weight readings, which is the natural limitation of the load cell.**

***Index Terms*—Indoor tracking, LeZi-Update, load sensors, probabilistic data association.**

Manuscript received March 14, 2006; revised September 22, 2006 and February 16, 2007. This paper was recommended for publication by Associate Editor C. Nugent and Editor D. Meldrum upon evaluation of the reviewers' comments. This work was supported in part by the National Science Council under the Project NSC93-2752-E-002-007-PAE, in part by the Excellent Research Projects of National Taiwan University under Project 95R0062-AE00-05, and in part by Intel under the Digital Health Project. This paper was presented in part at the IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, October 8–11, 2006.

W.-H. Liau and C.-L. Wu are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: fema@sgi.csie.ntu.edu.tw; f89922042@ntu.edu.tw).

L.-C. Fu is with both the Department of Computer Science and Information Engineering and the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: lichen@ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2007.911671

I. INTRODUCTION

THE GROWING power of the computer technologies have always been helping humans to realize many dreams, and one of them is extensive home automation, which usually involves integrated control of multiple home appliances. However, to make inhabitants feel more comfortable and to make life more convenient, the traditional Home Automation System (HAS) should evolve to the next stage. This stage would be to provide "context-aware" services actively, which can be deemed as providing appropriate services to the right people at the right moments and at the right places. Conceivably, the HAS is the core of a "smart home," a kind of context-aware environment with several kinds of sensors and controllable actuators. The sensors will extract information from the environment so that the services likely needed by the inhabitants are inferred, and finally, the confirmed services will be provided to the inhabitants through the corresponding actuators in the HAS. Among all sorts of context information to be collected in order for the HAS to do useful and relevant inferences, the locations are the most valuable one since the inhabitants would normally require different services based on which location in the home they are at. Therefore, in this study, we have focused on how to retrieve the inhabitants' location in a smart home.

To determine where an inhabitant is, several approaches can be adopted. Intrusive tracking approaches tend to make people uncomfortable, and some nonintrusive approaches also suffer from some inevitable drawbacks, which are detailed in Section II. In order to provide appropriate services while maintaining comfort, convenience, and nonintrusiveness, we adopt *sensory floor* as our tracking approach. Sensory floor can detect the pressure exerted on it, and here we utilize a number of load sensors to construct such a floor in a smart home so that human body weights can be collected nonintrusively. In addition to this hardware setup, we develop an algorithm to analyze the weight readings from the sensory floor whereby we can also keep track of the movement history of the inhabitants, as well as predict their possible movements. So the HAS can know where the inhabitants are after they walk into the home environment and step on the sensory floor within the smart home.

The main contributions of our system are listed next.

- *No need for inhabitants to wear any devices*: The inhabitants can be tracked by our system without carrying or wearing any additional devices in the home.
- *Tracking of multiple inhabitants*: Our system uses raw data returned from sensors to build a personal profile and uses this profile to track multiple inhabitants who are moving in the home concurrently.

- *Tolerance for noise in a cluttered home environment*: Our system tolerates noise from the sensors or from other inhabitants, especially when one track crosses another one, a phenomenon called cross-walking problem in this study.

The rest of this paper is organized as follows. In Section II, we introduce some related work, and a description of our system hardware is in Section III. Our inhabitants tracking algorithm is presented in Section IV, and experiment results are given in Section V. Finally, conclusions are drawn and some future works are discussed in Section VI.

II. Related Works

There are several related approaches to determine indoor location in a smart environment.

A. RADAR [5], [6]

This system utilizes the IEEE 802.11 based RF data network to obtain tracking signals. This system needs a trial calibration to develop a Radio Map, and then determines the location by finding the nearest signal characteristics in the Radio Map. Its error distance is 2.94 m for the median.

B. Cricket [1], [7], [8]

Cricket combines RF and ultrasound technologies to determine the location of a mobile device. The distance between the device and a beacon mounted on a wall can be calculated based on the difference in propagation speed between the RF and the ultrasound. Provided appropriate coordinate configuration of each beacon and enough information from beacons, Cricket can provide distance ranging and positioning precision of between 1 and 3 cm.

C. Active Badge [3]

Active Badge employs a low-cost, low-power infrared beacon-emitting device, which is designed to be worn by a person in an indoor environment. An IR signal with a unique ID is emitted from the badge worn by the person to the receiver grid attached to the ceiling, and then the location of the person can be calculated by the central server. It has good accuracy, but its precision is only at room level.

D. Active Bat [4]

Each Active Bat device emits an ultrasonic signal pulse, and the signal is received by sensors mounted at well-defined reference locations on the ceiling. The distance between the device and each sensor is calculated by measuring the time to flight of the ultrasonic signal in the central server. This calculation needs precisely synchronized clocks [28] operating between the device and the sensors. Provided sensor location map configured in advance, 95% of readings are within 9 cm of their true positions.

E. LANDMARC [9]

The LANDMARC uses a mobile device with Active RFID tags to emit a signal with an ID to receivers mounted on the walls or the ceiling in the smart environment. The propagation of RF signals varies, due to rapid change of the indoor environment. In order to overcome this problem, the LANDMARC uses some active tags that were deployed in the smart environment as reference landmarks. The location of the mobile device is calculated by finding the landmark which has the closest result to that of the mobile device. The error distance is around 1 m for the median.

F. EasyLiving [2], [10]

The EasyLiving project uses two stereo cameras to keep track of a user in the smart environment. The cameras can return the distance of some object in sight, and the distances are used in trilateration to calculate the location of the inhabitant. The tracking result is good, but its performance will be degraded when there is occlusion, which is caused by the image of a person covered by furniture or another person.

G. Sensory Floor

The Active Floor [23] is a kind of sophisticated load sensor, which can measure the Ground Reaction Force (GRF). The authors of [11] and [25] constructed smart floors by putting several Active Floors in some necessary positions. By analyzing the data from each Active Floor, they can use ten characteristics to indicate who each inhabitant is and what he wants, and the location of the inhabitant is confirmed while he/she is standing on the smart floor. However, it can not provide enough information to applications due to the limited number of sensors. The Ubiquitous Home [24] also uses floor pressure sensors to track inhabitants, but it cannot identify or reidentify persons after their paths cross since its sensors are binary detection units that lack weight measurement. The approach of smart floor tiles in [27], deployed at Gator Tech Smart House developed by the University of Florida, is also a kind of sensory floor. It is inexpensive and has high hit rate (between 87% and 81%) when tracking people. The above sensory floors, except for the smart floor tiles, do not provide a flat surface, like wooden floor covering the sensors, on which users can walk, which is more realistic and user-friendly. In addition, all of them do not solve the cross-walking problem or predict user movement.

Except for the last two approaches, all the above ones need devices worn by the inhabitants. However, inhabitants may forget to carry (or wear) such devices with them, or be unwilling to do so because these devices are intrusive or are uncomfortable and inconvenient. As for the stereo cameras, its performance will be degraded when occlusions happen. Therefore, we adopt sensory floor. The inhabitants need not to wear any device and there is no occlusion problem. Comparing with other sensory floor based approaches, we provide a flat walking surface, and developed an algorithm to track users, as well as predict their movement, thus solving the cross-walking problem which has not been solved by others.

III. System Hardware

In this section, we introduce the hardware of our tracking system, consisting of *sensory blocks*, Analog-to-Digital Converter (ADC) Module, and a server, as shown in Fig. 1.

A. Sensory Block

A sensory block is a basic unit measuring the exerted pressure, and it consists of two metal shelves and a load sensor. A

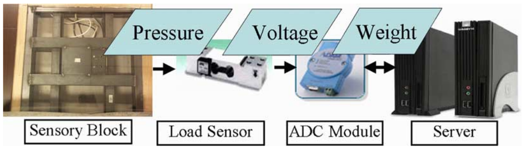

Fig. 1. Pressure exerted on a sensory block results in change of voltage at the load sensor. The ADC module digitizes this change and replies when queried by the server.

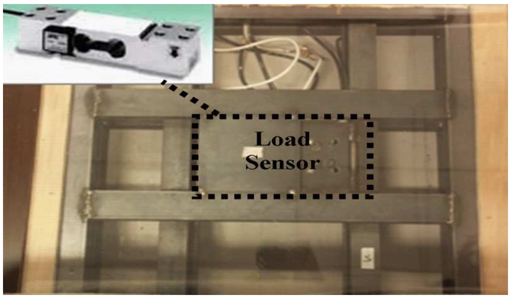

Fig. 2. Bird's eye view of a sensory block. The load sensor, located in the middle of the sensory block, is shown in the top left corner. A wooden plate is deployed above the sensory block to provide a flat surface.

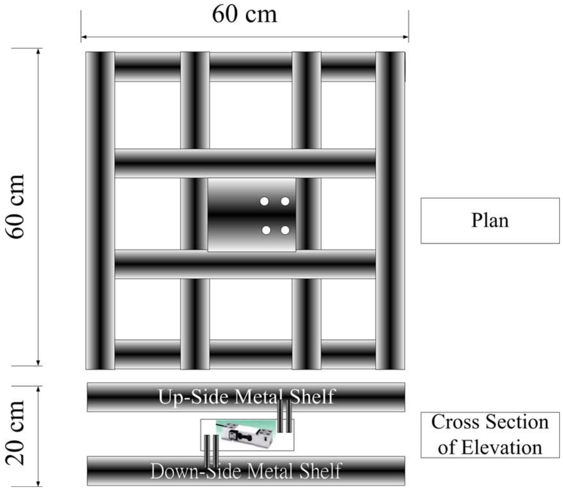

Fig. 3. Plan and elevation of a sensory block.

bird's eye view of the sensory block is shown in Fig. 2. A sensory block is sensitive to pressure variances. Once it is stepped on by an inhabitant, it perceives the pressure and return raw pressure data immediately. The load sensor connects the top and bottom metal shelves and is located between these two shelves in a vacant space reserved for it. Each sensory block is 60 cm square, and the height of the floor is 20 cm (as shown in Fig. 3).

Electric power with constant voltage is provided to every load sensor. Each load sensor responds to differences in pressure exerted on the sensory block and produce corresponding differences in voltage, because the electric resistance of the loading sensor changes as pressure changes.

B. ADC Module

Because voltage fluctuation readings from the load sensor are analog, an ADC module, consisting of multiple ADCs and one data acquisition module, is interposed between the load sensor and the server to convert the voltage values from analog to digital, so that the server can further process the digitized information. Each ADC can handle multiple load sensors at the same time. It senses the voltage differentials of load sensors and digitizing them. When the data acquisition module is queried by the server, it collects digitized readings from ADCs and encodes them as a string for the server.

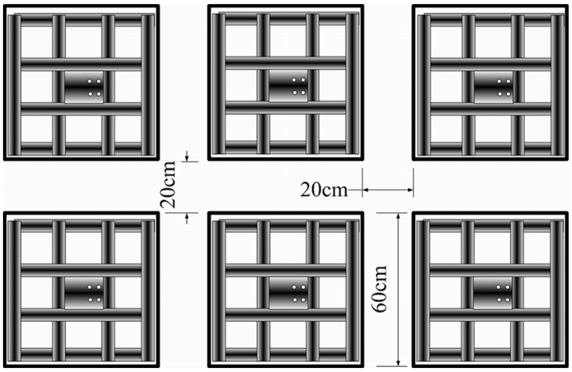

Fig. 4. The distance between two sensory blocks.

The resolution of the load sensor is 1% (5 Kg). According to our experimental observation, the weight reading is sometimes positive, even though there are no objects on the sensory block. This kind of false-positive reading is caused by an electromagnetic disturbance that affects the bare wires at the end of the connection between the load sensor and the ADC. The limited resolution and the electromagnetic disturbance produce some undesirable noise, which may influence the sensor reading slightly. This problem is discussed in Section IV-B.

C. Server

The raw data returned from the load sensors consists of real data and noise. The server sends requests to ADCs, parses returned strings, and then filters out noise. After that, the server infers the corresponding locations of the inhabitants and tracks them according to this data. The whole process is described in detail in Section IV.

D. Deployment and Configuration

The sensory floor in our experimental environment consists of 25 sensory blocks, and the distance between two blocks is 20 cm (as shown in Fig. 4). We spent 18 man-hours to deploy these sensory blocks, and another 4 man-hours to map them to physical locations for the configuration. In addition, since the bare surfaces of sensory blocks are unsuitable for walking, we cover them with conventional wooden flooring to provide a flat surface. However, the alignment of the wooden floor is not regular, so our tracking system encounters unpredictable noise caused by irregularities in the wooden floor installation. This problem is also discussed in Section IV-B.

IV. Inhabitant Tracking Via Sensory Floor

In this section, we first give a brief overview about the requirements analysis and the design concept. After this brief overview, we detail each process in the system flow.

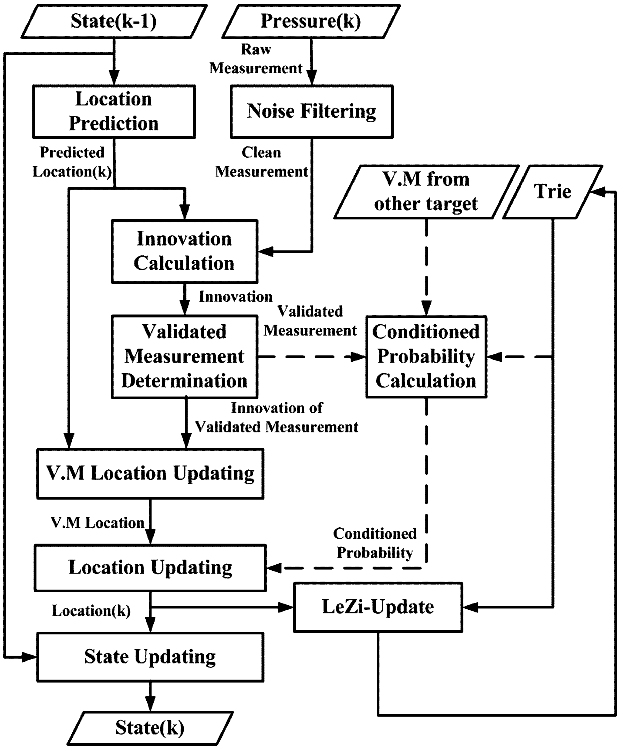

Fig. 5. Flow chart of LeZi-PDA.

In our environment, the structure of the wooden floor covering the load sensors produces clutter, which means irregularly producing different weight readings over multiple sensory blocks when the inhabitant only stepping on one of them. To solve these problems, we adopt the Probability Data Association (PDA) algorithm [13], [14], which is based on the Kalman filter [12] to track inhabitants. However, PDA does not consider another cluttered situation, which measurements are influenced with by other targets of interest, or are affected by interference, as in the crossover of two or more trajectories. Thus, we integrate the PDA algorithm with LeZi-Update [15], [16] and Active LeZi (ALZ) [17] to determine the conditional probability for each measurement. Due to the page limitation, more details about the requirement analysis and the design concept can be found in our previous work [29].

With the integration of the PDA and LeZi-Update, the total process of tracking the movement of an inhabitant from raw sensory readings is called LeZi-PDA and is shown in the flow chart in Fig. 5. Each process is divided to the following parts.

A. Location Prediction

First of all, we predict the location of the inhabitant by utilizing the latest state of the inhabitant as in (1)

$$\hat{X}(k) = F \bullet \hat{X}(k-1) \tag{1}$$

where $\hat{X}(k)$ consists of the 2-D coordinate (x, y) and velocity (v_x, v_y) of an inhabitant, and F is the prediction matrix, typically chosen as $\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. We extract the velocity from the latest state and predict the next state.

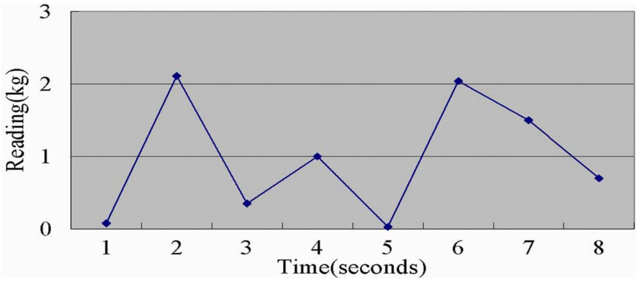

Fig. 6. Ambient noise for a sensory block.

B. Raw Measurement Collection and Noise Filtering

Some studies have pointed out that the mean normal walking speed for adult men is 1.34 m/s [18], and the sampling rate should be 2 Hz since average walking frequency is 1.75 Hz [26]. Because the sensory block has a size of 60 cm^2 and a sampling rate of 2 Hz, it is sufficient for human walking. Each sensory reading is converted from a voltage-based reading to a Kg-based reading by the server based on the specifications of the load sensor.

Based on our observations, there are two kinds of noise in the system, and they are described below.

- *Ambient Noise*: The end points of wires connecting the load sensor and ADCs are bare, and the electric wires are very long, so they receive electromagnetic waves in the air, thus influencing the weight readings. The typical range of influence caused by this electromagnetic disturbance is 0–3 Kg (as shown in Fig. 6); therefore, we ignore all weight readings which are less than 3 Kg.
- *Floor Deformation Noise*: When the sensory blocks are first placed in the room, hidden beneath the wooden floor, we can collect reasonably accurate responses from the sensory blocks. However, after a few months, some sensory blocks respond with nonzero readings, even after inhabitants have moved away from these sensory blocks. This is due to deformation of the wooden floor, which is caused by the sensory blocks being frequently stepped on by the inhabitants. The deformed wooden floor tiles do not readily spring back to their original uncompressed state when being stepped. However, a depressed wooden floor tile may restore to an uncompressed state after being stepped again. Weight readings with deformation noise are shown in Fig. 7, whereas the normal weight readings without deformation noise are shown in Fig. 9. Sometimes, floor deformation noise causes the reading noise to be even greater than 50 Kg. This seriously affects our inhabitant tracking results, but fortunately, this noise usually induces the same reading for a long duration as the furniture. Therefore, we can deal with this kind of noise with background model learning technique in computer vision. At first, this noised reading will produce the illusion that some inhabitant is there, but after a few seconds, if this noised reading still exists and maintain the same value, it will be deemed as background, just like some new furniture in the home.

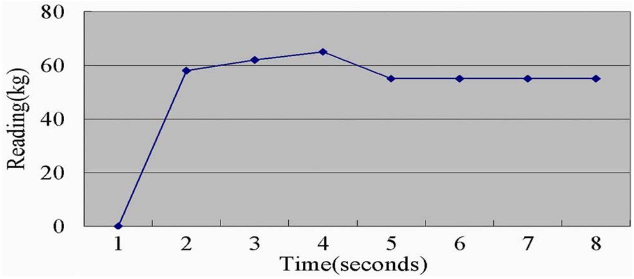

Fig. 7. Floor deformation noise.

All measurements, denoted as Y_i $(i = 1, 2, 3, \ldots, n_k)$, are filtered to eliminate the effects of noise and to produce clean measurements for later use.

C. Innovation Calculation

To calculate the innovation of each measurement is to calculate the error distance between each measurement and the predicted location. The innovation of a measurement is determined by the following equation:

$$v_i(k) = Y_i(k) - H \bullet \hat{X}(k|k-1). \tag{2}$$

In (2), $\hat{X}(k|k-1)$ is the predicted location, $Y_i(k)$ is the output of the ith clean measurement, and H is the observation matrix. Since the locations of the measurements can be determined accurately, H can be selected as an identity matrix.

D. Validated Measurement Determination

The validation gate (or validation region) is like a threshold, which enables the PDA algorithm to perform the tracking process efficiently and accurately by only considering validated measurements (measurements that fall within the validation gate). Therefore, given a reasonable value r, a measurement Y_i is a validated measurement only if its innovation v_i is less than or equal to some value according to the following equation:

$$\tilde{v}_k(\gamma) = \left\{Y_i : v_i^T S^{-1}(k) v_i \leq \gamma\right\} \tag{3}$$

where $S(k)$ can be obtained from the Kalman filter and other related information. The ith validated measurement is denoted as Z_i, and the entire set of validated measurements is denoted as $\overline{Z}(k) = \{Z_i | i = 1, 2, \ldots, m_k\}$.

E. Validated Measurement Location Updating

We use the following equation to update the location of each validated measurement:

$$\hat{Z}_i(k|k) = \hat{X}(k|k-1) + K(k) \bullet v_i(k). \tag{4}$$

In (4), $\hat{Z}_i(k|k)$ denotes the updated state of validated measurement Z_i, $K(k)$ is the standard Kalman gain, and $v_i(k)$ is the innovation of Z_i, which denotes the distance between the measurement Z_i and the predicted location.

The PDA filter will use the updated state $(\hat{Z}_i(k|k))$ of each validated measurement to update the predicted location later in the process.

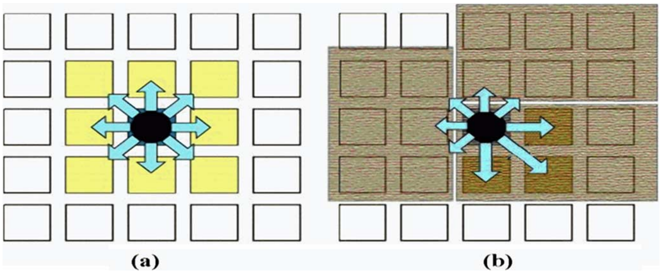

Fig. 8. (a) Load cell with uniform pressure distribution. (b) Load cell with directional pressure distribution.

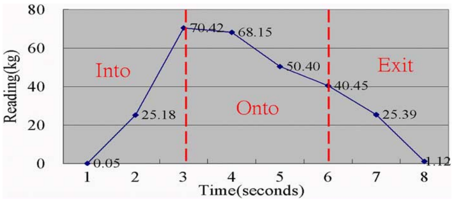

Fig. 9. Single sensory block weight reading trend.

F. Measurement Associated Probability

Theoretically, the pressure exerted on the floor will be distributed uniformly. However, because of the real physical configuration of the wooden floor, the pressure distribution will follow the alignment of the wooden floor. For example, if an inhabitant stands on the center block in the floor [see Fig. 8(a)], the eight blocks surrounding the center one will receive the same force. However, in reality, the right-bottom block and the right block in the figure will receive greater force than the other six blocks if the wooden floor is installed, as shown in Fig. 8(b). Therefore, we cannot mathematically model the real pressure distribution, so we introduce a set of physical phenomena to model the force distribution for determining the associated probability of each validated measurement.

The authors in [11] indicated that the Ground Reaction Force (GRF) can help identify a user and predict the actions the user is going to do. We also use the characteristics of the GRF diagram to analyze the activities of the inhabitant. Before the inhabitant walks on a sensory block, the sensory block responds zero reading to our system. While the inhabitant stands on the block, the pressure reading increases rapidly. Soon after, the user moves off the sensory block, and the pressure reading decreases gradually. Therefore, when a single inhabitant steps on a single block and then leaves, the change of the pressure reading with time is shown in Fig. 9.

Our system divides the weight reading shown in Fig. 9 into three parts: "**Into**," "**Onto**," and "**Exit**." These three parts are sequential.

- *Into* (shown in the left part of Fig. 9).
  If the weight reading of a sensory block shows the "Into" phenomenon, we declare that some heavy object is getting close to this sensory block. In this situation, we can say that an inhabitant is getting closer and closer to this sensory block.

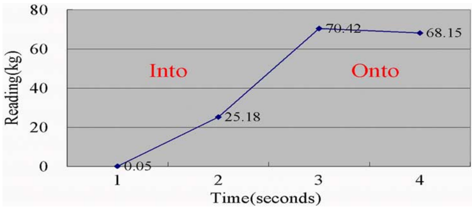

Fig. 10. Candidate block A.

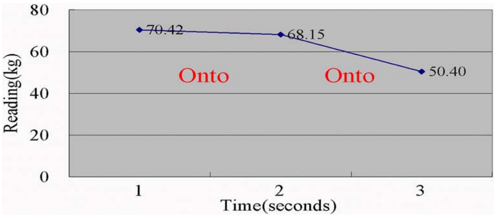

Fig. 11. Candidate block B.

- *Onto* (shown in the middle part of Fig. 9).
  This part follows the "**Into**" curve and shows that the heavy object is staying on this sensory block for a while. We conclude that the inhabitant is standing (or sitting) on this sensory block now.
- *Exit* (shown in the right part of Fig. 9).
  When the heavy object quits the sensory block, the weight reading shows this. This means that the inhabitant has left this block. However, the curve drops off gently because of the wooden floor deformation and the slow recovery.

Based on the time profile of the weight reading, three types of candidate blocks on which inhabitants might concurrently step are defined below.

- **Candidate block type A: Into + Onto**.
  The first kind of candidate block is characterized by readings exhibiting the "**Onto**" phenomenon, following the "**Into**" phenomenon, as shown in Fig. 10. It indicates that the inhabitant is getting close to the block or is standing on the block.
- **Candidate block type B: Onto + (No Exit)**.
  The second kind of candidate block is characterized by readings exhibiting the "**Onto**" phenomenon but no "**Exit**" phenomenon, as shown in Fig. 11. It shows that the inhabitant has stood on the block for a whole already and has not left the block yet.
- **Candidate block type C: Mixed**.
  The third kind of candidate block appears when two or more inhabitants meet together and step either on the same block or on neighboring blocks. It has no obvious curve phenomenon, so that our system cannot identify it by the trending models, as shown in Fig. 9. However, the reading of such a candidate block is generally greater than some weight threshold because of the total weight of the two (or more) inhabitants. Therefore, we consider those blocks whose readings over the threshold value (in our experiment, the threshold value was set to 20 Kg) as candidates. The curve is shown in Fig. 12.

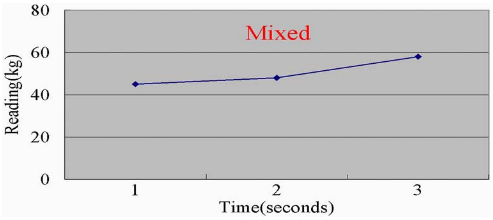

Fig. 12. Candidate block C.

This step marks all the sensory blocks that are probably being walked on by inhabitants as candidates. For different kinds of candidate blocks, we assign different associated probability values (between 0 and 1) for later use. Based on our experiment, the probabilities of the candidate blocks mentioned above are heuristically set as 0.6, 0.5, and 0.3, respectively.

G. Location Updating

After collecting the state of validated measurements and the associated probability (β_i) of each validated measurement, we can update the predicted location $\hat{X}(k)$ with the weighted sum of all validated measurements by using the following equation:

$$\hat{X}(k|k) = \frac{\left(\sum_{i=1}^{m_k} \hat{Z}_i(k|k) * \beta_i\right)}{\left(\sum_{i=1}^{m_k} \beta_i\right)}. \tag{5}$$

In (5), β_i denotes the associated probability of Z_i, and m_k is the number of validated measurements within the validation gate.

H. State Updating

The predicted location is updated in the last process. We use the last estimated state $\hat{X}(k-1)$ to calculate the velocity by means of the difference between the two locations, and then update the estimate $\hat{X}(k)$ using (6)

$$\hat{X}(k) = \begin{bmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \frac{\hat{x}(k)-\hat{x}(k-1)}{\Delta t} \\ \frac{\hat{y}(k)-\hat{y}(k-1)}{\Delta t} \end{bmatrix}. \tag{6}$$

I. Conditional Probability Calculation

The processes mentioned above can be used to track movement of the inhabitants in a cluttered environment if the tracks of the inhabitants are not entangled. But to conquer the tracking crossover, we adopt LeZi-Update to determine the conditional probability association as mentioned earlier. The flow chart is shown in Fig. 13.

- Location Prediction: First, those validated measurements belonging to all the crossover targets are collected to originate a new set of validated measurements, $Z'(k) = \{Z_i | i = 1, 2, \ldots, l_k\}$. For example, in Fig. 14, there are four validated measurements belonging to Target 2 and two validated ones belonging to Target 1, and the

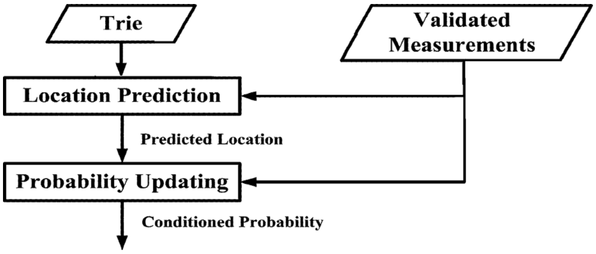

Fig. 13. Flow chart of conditional probability calculation.

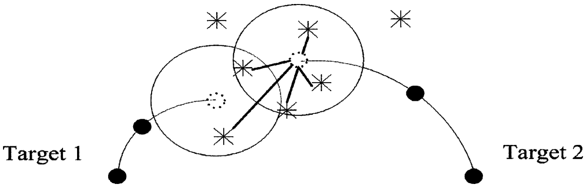

Fig. 14. Location prediction for target 2.

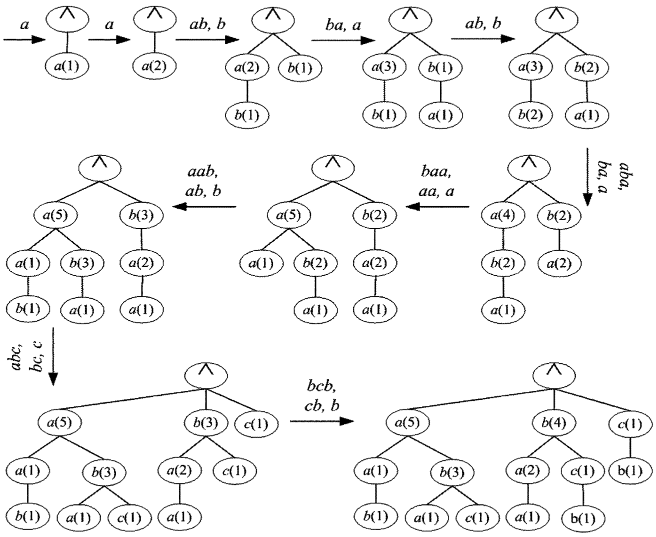

Fig. 15. An example of constructing a trie.

new set of validated measurements will consist of all five of them.

Second, we look up the trie previously constructed by the movement history of each target to find the probabilities of these five validated measurements, respectively, and later use the probability of each validated measurement as the associated weight.

For example, if the inhabitant has walked for a while and just moves from the sensory block whose "ID" is "a" to the sensory block whose "ID" is "b," the system will record the string "$\ldots ab$" as the movement phrase of the inhabitant. Taking at most two previous positions, i.e., "ab," into consideration, the LeZi probability $(L_i(c))$ of inhabitant i moving to sensory block "c" (the predicted location of the inhabitant is "c") will be $L_i(c) = (1/3) + (1/3) * ((1/4) + (1/4) * (1/10))$ according to (7), which is calculated based on the PPM exclusive strategy, if the trie is as the last one shown in Fig. 15

$$L(a) = p_2^a + \left(1 - \sum_{x=a,b,\ldots} p_2^x\right)\left[p_1^a + \left(1 - \sum_{x=a,b,\ldots} p_1^x\right)(p_0^a)\right]. \quad (7)$$

In (7), $L(a)$ denotes the probability of the appearance of the symbol "a" after perceiving two symbols, and p_i^x is the result of applying the order-i Markov model for some possible symbol through lookup of the trie.

Third, we predict the location of each target by calculating the weighted sum of the coordinates of these validated measurements

$$p_j(k) = \frac{\left(\sum_{i=1}^{l_k} L_j(Z_i(k)) * Z_i(k)\right)}{\left(\sum_{i=1}^{l_k} L_j(Z_i(k))\right)}. \quad (8)$$

In (8), $p_j(k)$ is the predicted position of target j at time instant k from looking up the history trie belonging to target j, $Z_i(k)$ is the coordinate of the ith validated measurement, and $L_j(Z_i(k))$ is the probability of the aforementioned coordinate for target j.

- Probability Updating: The pressure reading between two inhabitants can be deemed as the weight combination of these two inhabitants, and the weight combination is dependent on the distance relation in inverse proportion: the greater the distance between the sensor and the inhabitant, the lower the pressure reading conditioned by the inhabitant. We use the following equation to update the original probability associated with a validated measurement:

$$\alpha_{ji} = \beta_i * \varphi_{ji}, \text{ where } \varphi_{li} = \begin{cases} \frac{1}{d_{ji}}, & \text{else} \\ D, & d_{ji} = 0 \end{cases} \quad (9)$$

$$d_{ji} = \left|\hat{Z}_i(k) - p_j(k)\right| \quad (10)$$

where β_i is the original associated probability of the ith validated measurement, d_{ji} is the distance between the predicted location (determined from LeZi-Update) of the jth target and the location of the ith joint validated measurement, $j = 1, 2, 3 \ldots T$, and T is the number of crossing targets. Then, (5) can be modified as (11)

$$\hat{X}_j(k|k) = \frac{\sum_{i=1}^{m_k} \hat{Z}_i(k|k) * \alpha_{ji}}{\sum_{i=1}^{m_k} \alpha_{ji}}. \quad (11)$$

J. LeZi-Update

After the estimated location of an inhabitant is obtained, we update the trie for the inhabitant by using the standard procedure of the LeZi-Update algorithm. We divide a sensory block into nine small zones, as shown in Fig. 16, where all of the estimated locations can be categorized into some zones. If two continuous estimated locations are in different zones but belong to the same sensory block, we conclude that the inhabitant is staying at the same sensory block.

For example, assume that sensory block a is divided into nine parts, a_1 to a_9, as shown in Fig. 16. If the inhabitant moves

a_1	a_2	a_3
a_4	a_5	a_6
a_7	a_8	a_9

Fig. 16. Nine zones in a sensory block.

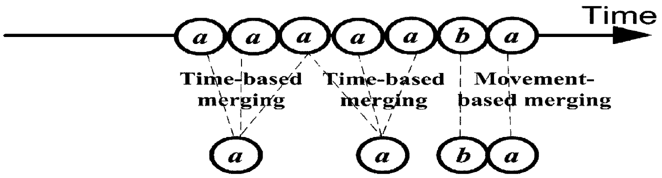

Fig. 17. Movement path example.

```
movement:={$x_1x_2x_3...x_n$}
add $x_1$ into the path
set pivot:=$x_1$
for i=2 to n
  if ($x_i$ == pivot) and (Time($x_i$)-Time(pivot)<threshold
          do nothing
  else
        add $x_i$ into the path
        pivot := $x_i$
  endIf
next
```

Fig. 18. Dynamic movement encoding procedure.

from a_i to a_j, where $1 \leqq i, j \leqq 9$ (for example, the inhabitant takes a small step on the sensory block), then we record the inhabitant's movement as $\{aa\}$. However, if the inhabitant moves from sensory block a to sensory block b, we record the inhabitant's movement as $\{ab\}$.

Our system needs to update the inhabitant's movement path dynamically because the system cannot receive his/her location directly due to the lack of any device worn by the inhabitant. There are, basically, three primitive dynamic path update schemes in the literature: distance-based [19], movement-based [20], and time-based [21].

The later two schemes seem quite suitable for our system because they do not need knowledge about the physical setting of the environment, but there are still some drawbacks arising from the nature of each scheme. For example, if the inhabitant's movement is $a \rightarrow b \rightarrow a$ between two updates, then the time-based update scheme will lose this activity. And if the update period is too short, the whole system will suffer from busy updating. On the other hand, the movement-based update scheme will lose the information about the duration of time during which the inhabitant steps on some sensory block.

Our system adopts the hybrid dynamic update scheme, as in [17]. This scheme has the advantage of being both movement-based one and time-based. Therefore, if the movement of the inhabitant is as shown in Fig. 17, then our system will update the path as $\{a, a, b, a\}$ by the procedure shown in Fig. 18.

If the encoded path is $\{a, a, b, a, b, a, a, b, c, b, \ldots\}$, it will be parsed as $\{a, a, (ab, b), (ba, a), (ab, b), (aba, ba, a), (baa, aa, a), (aab, ab, b), (abc, bc, c), (bcb, cb, b), \ldots\}$ according to ALZ algorithm [17]. Then, the movement dictionary will be constructed, as shown in Fig. 15, based on the parsed results and the decoding algorithm shown in Fig. 19.

```
initialize dictionary:=null
loop
    wait for next symbol from encoded path
    parse encoded path according to ALZ
    add new phrase to dictionary if necessary
    increment frequency for the parsed results
forever
```

Fig. 19. LeZi decoding algorithm.

K. Discussion

In our system, we do assume that only one person enters our environment at a time, and the time instants at which any two persons come in the environment, respectively, will be separated by at least some minimum time duration (so that enough initial distance between the two will be realized). Under this circumstance, the system will then be able to detect multiple people, and thus using multiple motion models for subsequent tracking. However, if two people enter the environment together (physically close) at the same time, then our system currently will not be able to tell whether it is two persons sticking together or not. Instead, our system will mistakenly treat the two as one person, but our system can still perform tracking. Later, after the two separate by some enough distance, our system can detect two readings and thus correctly recognize the two person situation and start another motion model to track the two. In addition, currently, our proposed algorithm is used to build the user's movement model, and therefore fails to consider the situations where people stay still for a long time. However, this problem can be easily solved by setting a threshold for motion string. In other words, when detecting that the user has stayed at some place for a period of time exceeding some time threshold, the system will stop inputting motion string into the movement model until the user starts to move again.

V. Experiment and System Evaluation

In this section, we will present the experiment results obtained with our inhabitant tracking system.

A. Single Inhabitant Tracking

Our first experiment was designed for single inhabitant tracking: We let the inhabitant walk around the sensory floor freely and precisely recorded the locations where he stayed.

The footprints of the inhabitant showed an inhabitant circling around along a dotted line on the wooden floor, as shown in Fig. 20(a), and the corresponding track estimated by our tracking algorithm is shown in Fig. 20(b). In Fig. 20(a), the numbers inside the circles, indicate the walking step sequences of the single inhabitant. On the other hand, the numbers inside the squares, show their corresponding estimated positions derived from our tracking system. Fig. 20(b) shows that the tracking algorithm performed better while the inhabitant was moving in a straight line than when the inhabitant changed direction. The twelfth step provides an obvious example of this phenomenon. Since our tracking algorithm is based on the

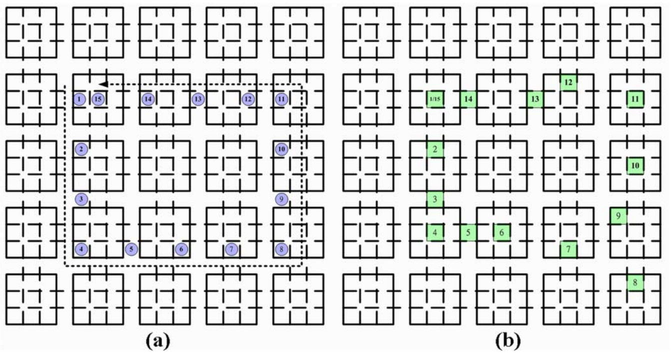

Fig. 20. (a) The recorded walking steps of a single inhabitant. (b) The estimates obtained with the tracking algorithm.

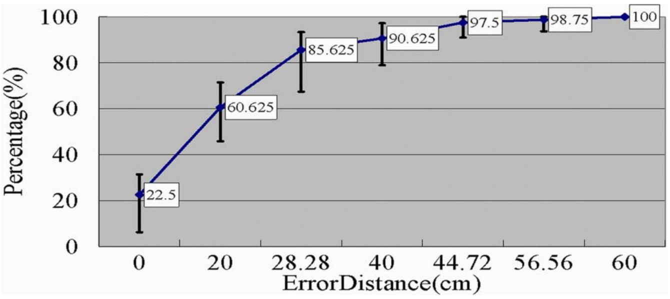

Fig. 21. The CDF of the error distance for a single inhabitant.

Kalman filter, the next location prediction is heavily dependent on the previous location and velocity; thus tracking may deviate more or less from the original route and move in a different direction.

Overall, the estimated track follows the real track of the inhabitant. However, the whole estimated track seems to be shifted one unit to the right. This is because the resolution of our load cell was greater than 20 cm. If the inhabitant is standing on units other than the center unit of one sensory block, our tracking system cannot differentiate them. The error rate of such movements will be higher than in more typical cases. On the other hand, if the inhabitant walks on the middle units between two nearby sensory blocks, our tracking system can track the inhabitant accurately because our approach is based on calculating the weighted sum of all validated measurements. Thus, the more validated measurements we get, the more accurate the estimate made by the system.

The CDF of error distances between the track estimate and the real movement is shown in Fig. 21, and 85% of the errors are less than 28.28 cm. According to data from [22], the average length of an adult foot was taken to be approximately 24 cm. Thus, the resolution of our inhabitant tracking algorithm is acceptable when applied to a home environment.

Another interesting observation from Figs. 22 and 23 is that the range of the pressure spread out along the X axis is greater than that along the Y axis. This is because the load sensors are covered by the wooden floor, which will cause signal distortion and make the pressure distribution irregular. Although we have used the validation gate and candidate blocks to alleviate the effects of the alignment (the installation direction) of the wooden floor, it still causes some side effects. From our observations, we learned that the range of the pressure spread out along the X axis was greater than that along the Y axis, and this can help us to build the noise model matrix Q and the $S(k)$, which are used in the Kalman filter.

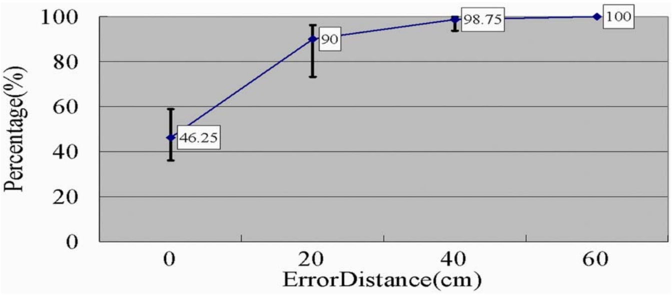

Fig. 22. X axis error distance (a single inhabitant).

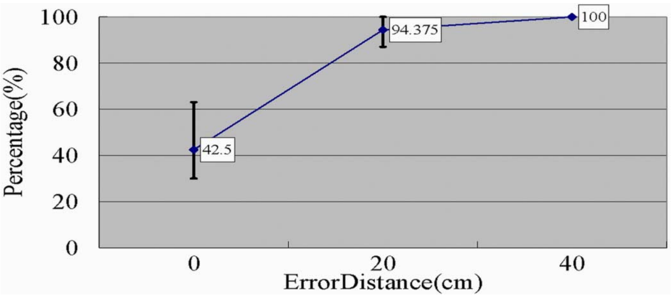

Fig. 23. Y axis error distance (a single inhabitant).

TABLE I
THE PROCESS TIME OF EACH PART OF THE TRACKING PROCESS

Process	Computing Time
Raw Data Collecting	<460ms
Inhabitant Tracking	10ms
LeZi Belief Weight Updating	10ms
Total Time	<500ms

In the experiment, we used a desktop PC (Intel Pentium III 1G and 384 MB of RAM) to implemen the inhabitant tracking algorithm. The average process time from initial clean measurement determination to the final state estimation of the inhabitant (the PDA process) was less than 10 ms. By summing the time spent on raw data collection (approximately 460 ms), the locations of all the inhabitants could be determined within 500 ms, which is short enough for general home use. We list the detailed results in Table I.

B. Multiple Inhabitant Tracking

In this experiment, there were two people. The weight of inhabitant #1 was 90 Kg, and that of inhabitant #2 was 50 Kg. The movement of inhabitant #1 was as shown in Fig. 20 (going in a straight line and then turning). Meanwhile, inhabitant #2 walked around randomly in a small area (in contrast to inhabitant #1). The CDF curves for inhabitant tracking are shown in Fig. 24. In this experiment, only 76.25% of the errors were less than 28.28 cm. The first reason may be that the pressure distribution over the wooden floor affected each inhabitant's validated measurement, and the second reason may be that the irregular

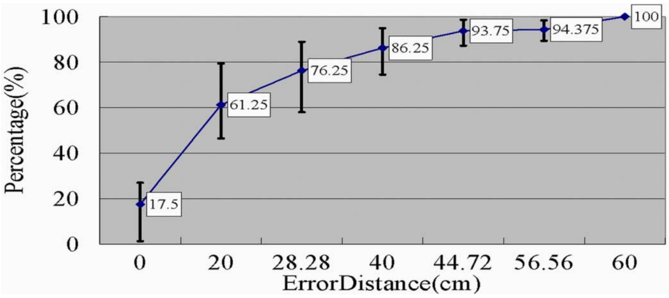

Fig. 24. The CDF of the error distance for two inhabitants.

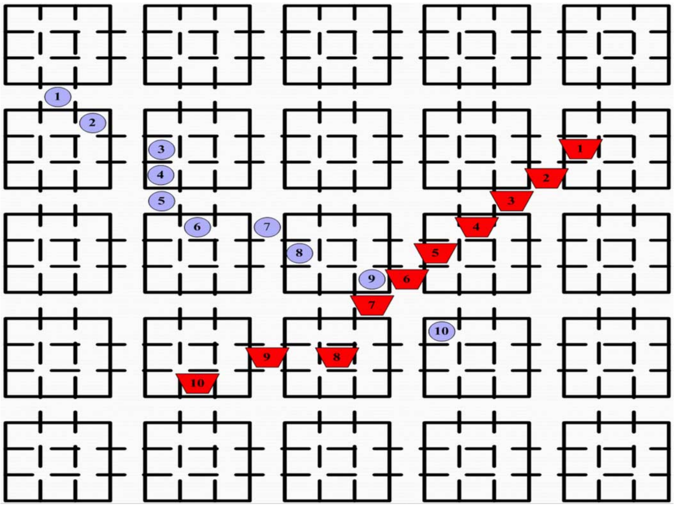

Fig. 25. Movement of two inhabitants as their paths cross each other.

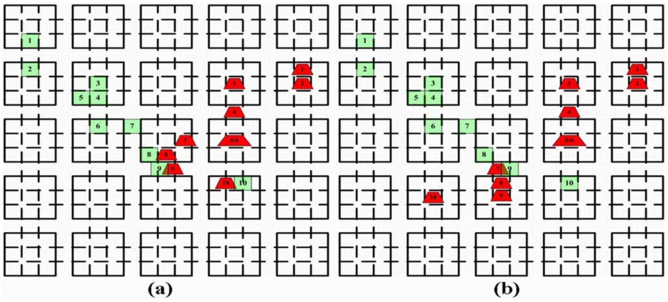

Fig. 26. (a) The results obtained with PDA. (b) The result obtained with LeZi-PDA.

movements of inhabitant #2 increased the noise for PDA calculation. The processing time was again less than 600 ms, because tracking two inhabitants involves the same process as tracking a single inhabitant with only the addition of the PDA process.

C. Multiple Inhabitants Tracking—Crossover

The third experiment focused on the cross-walking problem. We asked two people to walk along predefined paths (as shown in Fig. 25), which did cross each other to find out what LeZi-PDA could do in this situation. In Fig. 25, the numbers inside the circles, represent the movements of inhabitant #1, and the numbers inside the trapezoids, represent the movements of inhabitant #2.

First, we used the PDA to track these two inhabitants. The results shown in Fig. 26(a) reveal that the track of inhabitant #2 was dominated by the track of inhabitant #1. This is because the heavier inhabitant often produced more validated measurements, thus resulting in misleading readings for the lighter inhabitant. The conditional probability for each inhabitant associated with each validated measurement could alleviate this phenomenon, and the results produced by the LeZi-PDA tracking algorithm are shown in Fig. 26(b). In Fig. 26(b), clearly, there is an entanglement, in which the tracks of the two inhabitants cross each other. However, after the validated measurements were separated into two groups, LeZi-Update could highlight the validated measurements we needed to pay more attention to. By using LeZi-PDA, we could easily differentiate the tracks of the two inhabitants.

The average process time for LeZi-PDA was a little longer than that of PDA, because it needed more computations to look up the movement tries of each inhabitant and calculate the conditional probabilities associated with the validated measurements. The process time for LeZi-Update and calculation of the conditional probability was both less than 10 ms, and the average processing time was less than 500 ms.

VI. Conclusion and Future Work

One major objective with our smart home is to provide proper services to inhabitants at proper times and in proper places; thus tracking inhabitants is important. In this study, we utilized sensory blocks covered by wooden flooring to track inhabitants in a smart environment. To solve the problems encountered with our sensory floor, which were deformation of the wooden floor and the irregular weight distribution, confirmed by observing the pressure distribution and GRF, we adopted PDA and LeZi-Update to develop a tracking algorithm that can track multiple inhabitants in a nonintrusive way. The validation gate of PDA helps to filter out unreasonable measurements, and LeZi-Update provides the conditional probability for each validated measurements to separate crossing tracks caused by two or more inhabitants. We have presented a theoretic development and experimental results on tracking inhabitants in our smart home environment. In our extensive experiments, the error distance between the estimated location and the real location of an inhabitant is shown to be less than 28.28 cm (about an adult's foot size) in most situations (85%), and moreover the cross-walking problem has been solved.

Overall speaking, our LeZi-PDA tracking system has the following merits.

1) It utilizes the weights of inhabitants to achieve their nonintrusive tracking, who do not need to wear any gadgets.
2) It utilizes motion model to predict user movement, and it can track the randomly moving inhabitants in a cluttered environment with acceptable accuracy and efficiency.
3) It applies LeZi-Update with PDA to solve the cross-walking problem.

In our experiment, LeZi-PDA succeeded in most cases but might fail to work in some very rare situations (2 failures out of 25 experiments). To alleviate the problem caused by signal distortion from environmental noise, we plan to use smart floor tiles in [27] to cooperate with our system. Smart floor tiles can provide inexpensive solution with high accuracy, whereas our system can predict the user's movement. In addition, high accuracy when tracking multiple inhabitants is also a problem to be tackled in the near future. We aim to achieve this by incorporating more sensor inputs, such as computer vision from camera or the RFID from delicate badges, through the use of effective fusion methods. We are also investigating combining sensory floor with wireless sensor network to reduce the deployment work and the configuration time. Some other algorithms like hierarchical Bayesian network will also be explored to enhance our tracking performance.

References

[1] B. P. Nissanka, C. Anit, and B. Hari, "The cricket location-support system," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Networking*, 2000, pp. 32–43.

[2] Microsoft research center: EasyLiving [Online]. Available: http://research.microsoft.com/easyliving

[3] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91–102, Jan. 2, 1992.

[4] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," in *Proc. 5th Annu. ACM/IEEE Int. Conf. Mobile Comput. Networking*, 1999, pp. 59–68.

[5] P. Bahl, A. Balachandran, and V. Padmanabhan, "Enhancements to the RADAR user location and tracking system," *Microsoft Res. Tech. Rep.*, Feb. 2000 [Online]. Available: http://research.microsoft.com/~padmanab/papers/msr-tr-2000–12.pdf

[6] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. Conf. Comput. Commun.*, Mar. 2000, pp. 775–784.

[7] B. P. Nissanka, K. L. M. Allen, B. Hari, and T. Seth, "The cricket compass for context-aware mobile applications," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Networking*, Rome, Italy, 2001, pp. 1–14.

[8] S. Adam, B. Hari, G. Michel, and P. Nissanka, "Tracking moving devices with the cricket location system," in *Proc. 2nd Int. Conf. Mobile Systems, Applications, Services*, Boston, MA, 2004, pp. 190–202.

[9] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun.*, 2003, pp. 407–415.

[10] K. John, H. Steve, M. Brian, B. Barry, H. Michael, and S. Steve, "Multi-camera multi-person tracking for easyliving," in *Proc. 3rd IEEE Int. Workshop Visual Surveillance*, 2000, pp. 3–10.

[11] H. Robert and C. Rupert, "Recognizing movements from the ground reaction force," in *Proc. Workshop Perceptive User Interfaces*, Orlando, FL, , 2001, pp. 1–8.

[12] A. Gelb*, Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.

[13] Y. Bar-Shalom and T. E. Fortman*, Tracking and Data Association*. New York: Academic, 1987.

[14] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, 1975.

[15] B. Amiya and K. D. Sajal, "LeZi-update: An information-theoretic approach to track mobile users in PCS networks," in *Proc. 5th Annu. ACM/IEEE Int. Conf. Mobile Comput. Networking*, 1999, pp. 1–12.

[16] B. Amiya and K. D. Sajal, "LeZi-update: An information-theoretic framework for personal mobility tracking in PCS networks," *ACM/Kluwer J. Wireless Networks*, vol. 8, no. 2/3, pp. 121–135, Mar.–May 2002.

[17] K. Gopalratnam and D. J. Cook, "Active lezi: An incremental parsing algorithm for device usage prediction in the smart home," in *Proc. Florida Artificial Intell. Res. Symp.*, 2003 [Online]. Available: http://www-cse.uta.edu/~cook/pubs/flairs03k.ps

[18] Researchers study the walking speeds of older pedestrians TransSafety Inc. [Online]. Available: http://www.usroads.com/journals/rej/9704/re970404.htm

[19] L. Ben and J. H. Zygmunt, "Predictive distance-based mobility management for multidimensional PCS networks," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 718–732, Oct. 2003.

[20] I. F. Akyildiz and J. S. M. Ho, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Trans. Networking*, vol. 4, no. 4, pp. 629–638, Aug. 1996.

[21] C. Rose, "Minimizing the average cost of paging and registration: A timer-based method," *Wireless Networks*, vol. 2, no. 2, pp. 109–116, Jun 1996.

[22] Foot-Wikipedia [Online]. Available: http://en.wikipedia.org/wiki/Feet

[23] M. D. Addlesee, A. Jones, F. Livesey, and F. Samaria, "The ORL active floor [sensor system]," *IEEE Personal Commun.*, vol. 4, no. 5, pp. 35–41, Oct. 1997.

[24] T. Yamazaki, "Ubiquitous home: Real-Life testbed for home context-aware service," in *Proceedings. 1st Int. Conf. Testbeds Res. Infrastructures For Dev. Networks Communities*, Trento, Italy, Feb. 2005, pp. 54–59.

[25] R. Orr and G. Abowd, "The smart floor: A mechanism for natural user identification and tracking," in *Proc. Conf. Human Factors Comput. Syst.*, Apr. 2000, pp. 275–276.

[26] H. Fujiyoshi and A. Lipton, "Real-time human motion analysis by image skeletonization," in *Proc. Workshop Application Comput. Vision*, Oct. 1998, pp. 15–21.

[27] Y. Kaddourah, J. King, and A. Helal, "Cost-precision tradeoffs in unencumbered floor-based indoor location tracking," in *Proc. 3rd. Int. Conf. Smart Homes Health Telematics*, Jul. 2005 [Online]. Available: http://www.harris.cise.ufl.edu/projects/publications/Youssef-ICOST-2005–Final.pdf

[28] J. Elson, "Time synchronization in wireless sensor networks," Ph.D. dissertation, University of California, , 2003.

[29] W. H. Liau, C. L. Wu, and L. C. Fu, "Inhabitants tracking in a cluttered home environment via floor load sensors," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 8–11, 2006, pp. 4368–4373.

Wen-Hau Liau received the B.S. degree in computer science from the Tamkang University, Taipei, Taiwan, R.O.C., in 1998, and the M.S. degree in computer science from the National Taiwan University, Taipei, in 2005.

His research interests include smart environments, intelligent spaces, and topics related to them.

Chao-Lin Wu (M'03) received the B.S. degree in industrial technology education from the National Taiwan Normal University, Taipei, Taiwan, R.O.C., in 1996. Currently, he is working towards the Ph.D. degree in computer science at the National Taiwan University, Taipei.

His research interests include smart homes, smart environments, intelligent spaces, context-aware technologies, intelligent agents, and topics related to them.

Li-Chen Fu (M'84–SM'94–F'04) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1981, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1985 and 1987, respectively.

Since 1987, he has been a Faculty Member with the Department of Electrical Engineering and the Department of Computer Science and Information Engineering, National Taiwan University, where he currently is a Professor. His research interests include robotics, flexible manufacturing systems scheduling, shop floor control, home automation, visual detection and tracking, e-commerce, and control theory and applications.

Dr. Fu is a Senior Member of the IEEE Robotics and Automation Society and the IEEE Automatic Control Society.