Incremental Motion-Based Location Recognition

Seon-Woo Lee and Kenji Mase
ATR Media Integration & Communications Research Laboratories
2-2-2, Seika-cho, Soraku-gun, Kyoto 619-0288 JAPAN
{senu, mase}@mic.atr.co.jp

Abstract

This paper describes a method to recognize the current location of a user using a few simple sensors. The proposed method is based on the integration of incremental motions (e.g., walking) over time to determine location transitions. To handle uncertainty, a fuzzy reasoning method is introduced. The paper also describes a prototype location recognition system that uses a set of inexpensive and wearable sensors, a bi-axial accelerometer and a digital compass. The system uses a reliable motion recognition method featuring only two directional acceleration measurements. The effectiveness of the proposed method is demonstrated by experiments at six locations in an indoor environment.

1. Introduction

Location awareness is a key functionality in many applications of wearable computers. Specifically, recognizing the current location of a user is an essential part of capturing the user’s various situations in daily life.

Since Global Positioning Systems (GPS) are unavailable in indoor situations, a variety of systems, sensors, and techniques have been studied for indoor use. Many of these partial solutions can be categorized into two approaches: use of active or passive markers. Active beacon systems [11, 4] have been found to be the most common alternative providing reliable location information with minimal processing.

Another approach is the use of a camera and natural or artificial passive markers. In comparison with the methods using artificial markers, the approaches using naturally occurring features can offer more flexibility and no modifications to the environment. Sturier et al. [12] suggested a location detection method that uses a Hidden Markov model (HMM) and forward and downward looking hat-mounted cameras. Aoki et al. [1] developed a positioning system that uses a forward looking hat-mounted camera and dynamic programming algorithm on a stand-alone PC. Clarkson et al. [3] suggested a similar system that uses a wearable camera and HMM algorithm to recognize a spatially-based user’s situation, for example, “entering or leaving an office”. Unlike the above systems all of which identify discrete events, a system that uses an omni-directional camera and a probabilistic algorithm to track the location of a user has also been proposed in [10].

Instead of a vision system, Golding et al. [5] used a set of wearable sensors: accelerometer, magnetometer, light detector, and temperature sensors, to recognize the location of a user. In our previous work [7], we suggested a hybrid method to track the location of a user continuously. The proposed method employs a hybrid position measurement method: dead-reckoning for relative measurements and an infrared-based beacon method for absolute measurements. To measure the incremental motion of a user, we focused on recognizing a motor activity, i.e., walking.

Some interesting works have also been done specifically on activity-recognition including walking: Ashbrook [2] presented a basic idea about the usability of walking detection for context awareness using an one-hand keyboard called “Twiddler”. Recognition methods using accelerometers capable of distinguishing various activities of a user (sitting, standing, walking, ascending/descending, etc.) were proposed in [6, 9]. Unlike the above methods, a recognition method [8] has also been suggested not only to classify user activities, but also to count steps like a pedometer.

In this paper, we propose a recognition method using inexpensive sensors and a simple algorithm to find the current location of a user via the integration of incremental motions. The presented method is based on the same relative measurements as our previous work, but implements not a geometric but topological approach. While a user walks from a priori known location to another location, the method attempts to determine the location transitions like other topological localization methods, even though it is intrinsically based on dead-reckoning (also called “odometry”). As a result, our approach is not based on geometrical descriptions, e.g., \((x, y)\) in a two-dimensional space, but based on verbal descriptions of path segments like motor activities, e.g.,
walk straight, then go down the stairway, and turn right.

The robust and reliable recognition of unit motions is very important for dead-reckoning based location recognition. In our previous work [8], a nearest neighborhood method was used as a main classification method. Its performance, however, was not satisfactory enough to distinguish between walking on level ground and walking upstairs. It showed that an average recognition ratio of “up” is 83% for one user. In this work, we suggest a modified method using simple fuzzy logic based inferences to improve the recognition ratio. We propose a fuzzy reasoning based sequence retrieval method from a data base to determine a location transition with a sequence of recognized walking behaviors. Even for an identical location transition, a sequence will have some variances because of the variance of one step size in free walking. Fuzzy reasoning is introduced to handle such uncertainty. To evaluate the proposed method, we show some experimental results with selected locations that are very often used in office environments.

2. System Description

2.1. Overview

The proposed location recognition system is composed of three function blocks: a “sensing” block (which starts at the bottom layer), a “unit motion recognizer” (which captures user motions incrementally), and a “location recognizer” (which ends at the top layer). The sensing block reads the data of sensors with a fixed sampling rate, and then executes a set of pre-processing including filtering and computing statistical properties. The “unit motion recognizer” block performs the recognition of the motor activities of the user. Specifically, our recognizer can detect a forward step, and discriminate between walking on level ground (“level”), going up (“up”) a stairway, or going down (“down”) a stairway. It can also count the number of steps. The proposed detection method was designed from an analysis of acceleration characteristics, and the classification method uses fuzzy logic based inferences. The “location recognizer” keeps track of the walking behaviors and heading measurements. Whenever a new unit motion is detected, the location recognizer tries to determine the current location of the user by a fuzzy logic based matching algorithm that uses the accumulated sequence of motions from a data base, which is built offline in the training phase.

Consequently, the presented system operates in two phases, i.e., a training phase and a recognition phase. In the training phase, the system records a sequence of unit motions while the user walks from one location to another location. Using the recorded sequences, we can easily build a fuzzy rule base, in which a location transition can be represented by one fuzzy rule, as described specifically in Section 2.3. In the recognition phase, the system continuously tries to find unit motions and to recognize a location transition from a known starting location. If it finds a matched sequence, it then changes the user’s current location. This process is repeated with a new starting location.

For experimental verification, we implemented our method on a prototype as shown in Figure 1. The prototype consists of a notebook PC (Intel Pentium-MMX, 266 MHz), a card type data acquisition module, and a sensing module. The body-worn sensing module consists of a bi-axial accelerometer (ADXL 202EB from Analog Devices Inc.) and a digital compass module (HMR-3000 from Honeywell Co.). The implemented sensing module is small and light, and is assumed to be fixed to the middle of the waist of the user. The reason for the selection of this position is to measure the forward and upward accelerations and heading accurately. In addition, comfort and unobstructed wearability are provided on the belt as evidenced by other mobile devices (e.g., cellular phones and pedometers).

![Figure 1. Use and inside view of sensing module](image)

The accelerometer measures the user’s forward and upward accelerations. The acceleration signals are low pass filtered via two cascaded filters: one is a hardware-type RC filter with a 1 [KHz] cut-off frequency, and the other is a second-order elliptic digital filter with a 5 [Hz] cut-off frequency. The data is read every 20 [msec], i.e., the sampling frequency is 50 [Hz].

The digital compass can give us the azimuth heading and roll, and the pitch information of the module via an RS-232 serial communication channel. Among these data, we only use the heading data being read every 100 [msec]. To reduce heading errors derived from body movements, a quantization technique is introduced.
2.2. Unit motion recognition

In our approach, the system has to recognize not only walking behaviors, but also count the number of steps. This means that the system should discriminate human walking in one cycle units. In ergonomics [13], one cycle of human walking (called the “gait cycle”) is generally defined in terms of an interval of time during which one sequence of regularly recurring succession of events is completed. These sequential events are as follows: foot strike, opposite toe-off, opposite foot strike, and toe-off. Instead of a whole gait cycle, we consider a half gait cycle including only one foot strike, since it is easier to detect the foot strike on an acceleration measuring based system.

Figure 2 shows the typical trajectories of two pre-processed accelerations when a user walks on level ground, goes up a stairway and goes down a stairway. As shown in the figure, we can easily find the half gait cycle of the user.

To discriminate one cycle of a step, the peak acceleration values can be used. By using a conventional peak detection algorithm, the system tries to find the positive and negative peak values of the upward acceleration, such as shown in Figure 2 with the mark o. From here, the peaks are denoted as $z_{peak}^+(t_1)$ and $z_{peak}^-(t_2)$, where $t_{1,2}$ is defined as the time interval from the moment of the last step detection to the current peak detection for each peaks.

When these two peaks are found, the system tests the following conditions to determine a new step.

1. Whether the absolute difference value of the two peaks is above a threshold value, i.e.,
$$|z_{peak}^+(t_1) - z_{peak}^-(t_2)| > Th_1,$$
2. Whether the standard deviation over 50 samples of the forward acceleration (notation: $\sigma_x(t)$) is above a threshold value, i.e., $\sigma_x(t) > Th_2$.
3. Whether the time interval between the two peaks is found in a specific duration, i.e.,
$$Th_3 < t_2 - t_1 < Th_4$$
where the parameters $Th_1$ and $Th_2$ represent minimum variation of acceleration for a walking behavior. The $Th_3$ and $Th_4$ are introduced to prevent false detection from other body movements such as standing or sitting. Each values, therefore, can be determined easily from the statistical properties of the obtained data in learning phase. If the above conditions are satisfied, the system performs the fuzzy reasoning with the current feature values and linguistically described fuzzy rules.

Fuzzy logic has been used in many applications [14], specifically because it can handle ambiguity in decision making for expert systems. We applied fuzzy reasoning in our work to determine a forward step and classify it among the walking behaviors considered: “level”, “down”, and “up”.

The inputs of the fuzzy reasoning process are selected as: a standard deviation over 50 samples of the forward acceleration and upward acceleration, which are denoted as $\sigma_x(t)$ and $\sigma_z(t)$, respectively, and a mean over 50 samples of the forward acceleration (notation $m_x(t)$). The input vector is defined with these three values as:
$$\vec{u}(t) = \{u_1, u_2, u_3\} \equiv \{\sigma_x(t), \sigma_z(t), m_x(t)\}. \quad (1)$$

We define the linguistic labels “Small (S)”, “Medium (M)”, “Large (L)”, and “Very Large (VL)” for the two inputs $u_1$ and $u_2$, and the labels “Negative (N)” and “Zero (Z)” for $u_3$. As shown in Figure 4, the membership functions of each linguistic label are defined as a Gaussian function:
$$\mu_i^{(u)}(u_k) = e^{-\frac{(u_k - m_i)^2}{\sigma_i^2}} \quad (2)$$
where the index $i$ represents the element of the input vector, i.e., $i = 1, 2, 3$, and the index $l$ can be one of the linguistic labels such as S, M, L, or VL for $i = 1, 2$, or N or Z for $i = 3$.

There are several advantages in using a Gaussian function as a membership function. First, we can easily adjust the characteristics of the linguistic labels with its parameters, i.e., the mean and standard deviation values denoted as $m_i^l$ and $\sigma_i^l$ in Eq. 2. Second, if it is possible to get stochastic properties such as the mean and variance, then we can use it in order to design the membership function. In our case, we can chose the parameters of the membership function from a set of sampled training data.

Using these linguistic labels, we build a fuzzy rule base for three walking behaviors as follows:
\[ R_L: \text{IF } \sigma_x(t) \text{ is M AND } \sigma_y(t) \text{ is S AND } m_x(t) \text{ is Z, OR } \sigma_x(t) \text{ is L AND } \sigma_y(t) \text{ is M AND } m_x(t) \text{ is Z, OR } \sigma_x(t) \text{ is VL AND } \sigma_y(t) \text{ is L AND } m_x(t) \text{ is Z, THEN the current step is “level”}. \]

\[ R_D: \text{IF } \sigma_x(t) \text{ is VL AND } \sigma_y(t) \text{ is VL, THEN the current step is “down”}. \]

\[ R_U: \text{IF } \sigma_x(t) \text{ is S AND } \sigma_y(t) \text{ is M AND } m_x(t) \text{ is N, OR } \sigma_x(t) \text{ is M AND } \sigma_y(t) \text{ is L AND } m_x(t) \text{ is N, THEN the current step is “up”}. \]

Note that the fuzzy rule for “level” \((R_L)\) is or-ed with three different rules, which can be considered as representations for slow, normal, and fast level walking, respectively.

Using the fuzzy rules and given input vector, we compute the truth values of each proposition as:

\[
\omega_l = \max\{\min(\mu_1^L(u_1), \mu_2^S(u_2), \mu_3^Z(u_3)), \min(\mu_1^L(u_1), \mu_2^S(u_2), \mu_3^Z(u_3)), \min(\mu_1^L(u_1), \mu_2^S(u_2), \mu_3^Z(u_3))\},
\]

\[
\omega_u = \min(\mu_1^L(u_1), \mu_2^S(u_2), \mu_3^Z(u_3)),
\]

\[
\omega_m = \max\{\min(\mu_1^L(u_1), \mu_2^S(u_2), \mu_3^Z(u_3)), \min(\mu_1^L(u_1), \mu_2^S(u_2), \mu_3^Z(u_3)), \min(\mu_1^L(u_1), \mu_2^S(u_2), \mu_3^Z(u_3))\}. \tag{3}
\]

Here, the \(\min\) and \(\max\) operations are used as “AND” and “OR” operations in the fuzzy rules.

If the maximum value of the three resulting truth values is greater than a threshold value, \(Th_5\), then the unit motion recognizer eventually determines the current step as one of the three walking behaviors. To improve the recognition ratio between the “down” and other behaviors, we also use the same feature as described in [8]. Figure 3 shows typical cross-correlation trajectories between forward and upward accelerations of three behaviors at the time a step is detected. As shown in Figure 3, the cross-correlation characteristic of “down” is very distinguishable from the others. From finding the lag indices which represent first peak and valley values, we can recognize easily the “down” behavior.

2.3. Location recognition

As discussed briefly in Section 2.1, the location recognizer tries to determine a location transition via the fuzzy reasoning of a recorded sequence of unit motions from a data base. The data base is composed of fuzzy rules for describing typical sequences while the user moves from one location to another location.

For example, let us consider the situation when a user gets a cup of coffee from a coffee maker, which is located at another place from his/her seat (or office). He/she first goes to the coffee area, gets a cup of coffee, and returns.
If the number of locations trained is $N$, then the fuzzy rule of a transition from location $i$ to location $j$ is represented by the following fuzzy rule:

$$R^j_i: \text{IF } P_1 \text{ is } M^{ij}_1 \text{ AND } P_2 \text{ is } M^{ij}_2 \text{ AND} \cdots \text{ AND } P_k \text{ is } M^{ij}_k, \text{ THEN the location of the user is changed from } i \text{ to } j, i, j = 1, \cdots, N, i \not= j$$

where $M^{ij}_k$ is a fuzzy number characterized by membership functions $\mu_{M^{ij}_k}(P_k)$, $k = 1, \cdots, 6$. The fuzzy number is defined as a fuzzy set, like the linguistic label, but conceptually it represents a number with fuzziness as shown in Figure 6.

From the training phase, we can easily calculate the stochastic features of each element, and then determine the Gaussian membership function for a transition by using the mean and variance values. In practice, the size of one step can be widely varied in free walking even for the same user. From our prior work [8], the step size at a normal speed has a variance of 20% with respect to slow and fast walking. Consequently, even for the same path, the number of steps also has such variances. We need to introduce a method capable of handling such vagueness.

When the unit motion recognizer detects a new step, the location recognizer updates this descriptor vector and computes the truth value of each rule with the current descriptor. Here, the min operation is also used for an “AND” operation for each rule as:

$$\omega^j_i(\hat{P}(t)) = \min_{k=1,\ldots,6} \mu_{M^{ij}_k}(P_k(t)), \quad (5)$$

where current descriptor vector $\hat{P}(t)$ is defined as a descriptor vector (see Eq. 4), and each element is the number of steps accumulated till now.

Finally, the location recognizer finds the maximum value among the calculated truth values, and compares it with a threshold value, $Th_a$. If the maximum value is greater than the threshold value, the system determines the current location of the user via the found transition and the known starting location. Once the location recognizer determines the current location, the path descriptor vector and the starting location are reset to zero and the changed location, respectively.

3. Experimental Results

We examined the performance of the proposed recognition method using the following parameters (threshold values):

$$Th_1=0.3 \ [g], Th_2=0.08 \ [g], Th_3=0.1 \ [\text{sec}],$$
$$Th_4=0.6 \ [\text{sec}], Th_5=0.5, Th_6=0.6.$$  

where the unit $[g]$ represents a gravitational acceleration, i.e., $1 \ [g] \approx 9.8 \ [m/sec^2]$ in normal condition. The threshold values were selected manually by analyzing the sampled data in learning phase. The threshold $Th_a$ and $Th_b$ determine a sensitivity of unit motion and location recognizer. Consequently, it is possible to adjust the recognition ratio with changing these values, but we should note that there is a trade-off between sensitivity and noise immunity.

For the unit motion recognizer, the Gaussian membership functions for the linguistic labels $S$, $M$, $L$, $VL$ and $N$, $Z$ were selected with the parameters, which are listed in Table 1 (also shown in Figure 4).

<table>
<thead>
<tr>
<th>$m_M^i$</th>
<th>$m_M^1$</th>
<th>$m_M^L$</th>
<th>$m_M^{VL}$</th>
<th>$\sigma_M^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09</td>
<td>0.11</td>
<td>0.13</td>
<td>0.15</td>
<td>0.06</td>
</tr>
<tr>
<td>$m_N^i$</td>
<td>$m_N^L$</td>
<td>$m_N^{VL}$</td>
<td>$\sigma_N^i$</td>
<td></td>
</tr>
<tr>
<td>0.13</td>
<td>0.2</td>
<td>0.27</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>$m_Z^i$</td>
<td>$\sigma_N$</td>
<td>$m_Z^L$</td>
<td>$\sigma_N^i$</td>
<td></td>
</tr>
<tr>
<td>-0.07</td>
<td>0.12</td>
<td>0.06</td>
<td>0.15</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. The mean and standard deviation values of membership functions for the unit motion recognizer

![Figure 4. Membership functions of linguistic labels for unit motion recognition](image)

At first, for the evaluation of the unit motion recognizer, the walking data of one person were collected. The sampled data employed had 1852 steps for “level” walking and 156 steps each for going “up” and “down” a stairway.

Table 2 shows average results of the unit motion recognition performance for the counting of steps as well as the classifying of walking behaviors. Although
Table 2. Recognition ratios of the unit motion recognizer

<table>
<thead>
<tr>
<th>Unit [%]</th>
<th>level</th>
<th>down</th>
<th>up</th>
</tr>
</thead>
<tbody>
<tr>
<td>level</td>
<td>98.1</td>
<td>0.2</td>
<td>1.7</td>
</tr>
<tr>
<td>down</td>
<td>1.9</td>
<td>98.1</td>
<td>0</td>
</tr>
<tr>
<td>up</td>
<td>8.3</td>
<td>0</td>
<td>91.7</td>
</tr>
</tbody>
</table>

the presented results are derived from an experiment on one user, the presented method can be easily adapted to other users by changing the parameters of the membership functions. Compared to our prior work [8], the proposed method shows a better performance for one specific use. We believe that the proposed method can also provide a better performance for many users. Further study should be performed to evaluate the generalization capability of the method.

To evaluate the location recognition, the following six locations were selected as shown in Figure 5:

- My seat (#0): can be considered as the center in office activities.
- Printer room (#1): has printers and copiers.
- Toilet (#2): is the most frequently used location for everyone.
- Coffee area (#3): has coffee machines and refrigerators.
- Experimental room (#4): is nicknamed as the “Live-house”.
- Library (#5): is located on the second floor (not shown in the figure).
- Cash machine (#6): is located on the first floor (not shown in the figure).

The selected locations may be considered as places often used in daily office activities. In the figure, the paths between two locations are denoted by blue solid lines.

In a daily office environment, almost all user activities are to go to certain locations for various purposes, and then to return to one’s seat. Based on this fact, we collected data while a user walked along round paths from location #0 to others #1~6 two times. From the data, we built a fuzzy rule base for location recognition as shown in Table 3. The listed paired numbers (denoted \( m/\sigma \)) represent values of a mean and standard deviation of a Gaussian function for a fuzzy number. For example, two fuzzy numbers characterized by \((2/3)\) and \((10/5)\) are plotted in Figure 6. The returning paths

Table 3. The fuzzy rule base for the location recognizer for the paths from #0 to #1~6, and #5,6 to #0

<table>
<thead>
<tr>
<th>path</th>
<th>( L_M )</th>
<th>( L_\alpha )</th>
<th>( L_\alpha )</th>
<th>( L_\alpha )</th>
<th>( D ) or ( U )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 → 1</td>
<td>(6/3)</td>
<td>(26/7)</td>
<td>(0/5)</td>
<td>(2/5)</td>
<td></td>
</tr>
<tr>
<td>0 → 2</td>
<td>(2/3)</td>
<td>(45/10)</td>
<td>(2/5)</td>
<td>(6/5)</td>
<td></td>
</tr>
<tr>
<td>0 → 3</td>
<td>(25/7)</td>
<td>(54/10)</td>
<td>(0/5)</td>
<td>(2/5)</td>
<td></td>
</tr>
<tr>
<td>0 → 4</td>
<td>(50/10)</td>
<td>(0/5)</td>
<td>(0/5)</td>
<td>(2/5)</td>
<td></td>
</tr>
<tr>
<td>0 → 5</td>
<td>(3/3)</td>
<td>(92/10)</td>
<td>(58/10)</td>
<td>(10/5)</td>
<td>U(25/5)</td>
</tr>
<tr>
<td>0 → 6</td>
<td>(4/3)</td>
<td>(102/10)</td>
<td>(10/5)</td>
<td>(4/5)</td>
<td>U(25/5)</td>
</tr>
<tr>
<td>5 → 0</td>
<td>(55/10)</td>
<td>(10/5)</td>
<td>(5/3)</td>
<td>(98/10)</td>
<td>U(25/8)</td>
</tr>
<tr>
<td>6 → 0</td>
<td>(3/3)</td>
<td>(3/3)</td>
<td>(116/15)</td>
<td>(110/15)</td>
<td>U(45/15)</td>
</tr>
</tbody>
</table>

Figure 5. Location map: shows the selected locations and paths

Figure 6. Membership functions of two fuzzy numbers: \((2/3)\) and \((10/5)\) as examples
from #1 to #0 can be represented easily by changing the elements to have opposite relationships, i.e., north $\leftrightarrow$ south and east $\leftrightarrow$ west.

Table 4 shows results of location recognition. As shown in the table, we could get promising results even with minimum hardware and processing power. The average recognition ratio for 12 location transitions is 86.7 [%].

<table>
<thead>
<tr>
<th>path</th>
<th># of visits</th>
<th># of failure</th>
<th>accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 $\rightarrow$ 1</td>
<td>14</td>
<td>1</td>
<td>93</td>
</tr>
<tr>
<td>1 $\rightarrow$ 0</td>
<td>14</td>
<td>1</td>
<td>93</td>
</tr>
<tr>
<td>0 $\rightarrow$ 2</td>
<td>13</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>2 $\rightarrow$ 0</td>
<td>12</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>0 $\rightarrow$ 3</td>
<td>15</td>
<td>2</td>
<td>85</td>
</tr>
<tr>
<td>3 $\rightarrow$ 0</td>
<td>14</td>
<td>2</td>
<td>83</td>
</tr>
<tr>
<td>0 $\rightarrow$ 4</td>
<td>8</td>
<td>1</td>
<td>88</td>
</tr>
<tr>
<td>4 $\rightarrow$ 0</td>
<td>8</td>
<td>1</td>
<td>88</td>
</tr>
<tr>
<td>0 $\rightarrow$ 5</td>
<td>7</td>
<td>1</td>
<td>86</td>
</tr>
<tr>
<td>5 $\rightarrow$ 0</td>
<td>7</td>
<td>1</td>
<td>86</td>
</tr>
<tr>
<td>0 $\rightarrow$ 6</td>
<td>5</td>
<td>1</td>
<td>80</td>
</tr>
<tr>
<td>6 $\rightarrow$ 0</td>
<td>5</td>
<td>2</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 4. The results of the location recognizer for 12 paths

The computational complexity of the proposed method was examined. Our prototype (Intel Pentium-MMX, 266 MHz) showed that average computing time for whole processing except sensor data reading is 0.0216 [ms] in the range of 0.018 to 0.026 [ms]. It took 8.8 [ms] approximately to read two acceleration data from the bi-axial accelerometer using the PCMCIA-type data acquisition module, and 2.5 [ms] to read the heading data from the digital compass module via a standard serial port. From this finding, the proposed method could be implemented easily on an embedded system.

From an analysis of the experimental results, we could see that the main error source originated from the mis-recognition between the “level” and “up” steps. Although the wrong detection of a unit motion can seriously affect to both cases of short and long paths, the effect on a long path including “up” behaviors like #6 $\rightarrow$ #0 is more serious.

We also found a limit in our method for the case of a long path. More specifically, the location recognizer shows the location transition before the user goes to the destination. This is because that the determination of the transition is based on simple accumulated numbers of steps instead of a whole sequence. This problem can be partially solved with a higher threshold value, $T_{thb}$, but a higher threshold value can lead to a lower recognition rate. We are currently studying how to implement a whole sequence to solve the problem.

4. Conclusion and Future Work

In this paper, we proposed an incremental motion based location recognition method using a few simple sensors. The proposed method is composed of three function blocks: a “sensing” block for reading and pre-processing raw data, a “unit motion recognizer” for detecting and classifying walking behaviors, and a “location recognizer” for determining location transitions via the fuzzy reasoning of the proposed path descriptor. The improved walking behavior method showed a classification ratio exceeding 90%. We also demonstrated the performance of the location recognition by experiments at six locations in an office environment.

The proposed location recognition is based on dead-reckoning intrinsically. It is clear that improving the unit motion recognition will enhance the performance of the location recognition. To improve the unit motion recognition, our future work includes adding different type of sensors, different positions on the body, and modifying the algorithm. We also plan to evaluate the generalization capability of the unit motion recognition method via experiments on many people. We are currently focusing on using a different type of path descriptor to enhance the location recognition performance. Our aim is to implement a smaller and lighter prototype than the current one, and to also give the prototype a wireless connection capability to central mobile units (PDAs or laptop PCs) for more comfortable usage.

Acknowledgments

This work has been supported partially by CREST of JST (Japan Science and Technology Corporation). We would like to thank the members of ATR Media Integration & Communications Research Laboratories, and their President Ryohei Nakatsu for their continuous support.

References


