Automated detection of wandering patterns in people with dementia

Nhu Khue Vuong BEng[a]
Syin Chan PhD[a]
Chiew Tong Lau PhD[a]

[a]School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore; E: VUON0004@e.ntu.edu.sg

N.K. Vuong, S. Chan, C.T. Lau. Automated detection of wandering patterns in people with dementia. Gerontechnology 2014;12(3):127-147; doi:10.4017/gt.2014.12.3.001.00 **Purpose** This study focuses on travel patterns of people with dementia (PWD), which can be classified as direct, pacing, lapping, or random based on the Martino-Saltzman (MS) model. **Method** We take the movement data of five nursing home residents with dementia, comprising 220 travel episodes of room-to-room movements, and manually applied MS model to classify the travel patterns in each episode. Next, we propose two approaches to automatically classify the travel patterns: a machine learning approach and a deterministic predefined tree-based algorithm. In the machine learning approach, eight classical algorithms including Naïve Bayes (NB), Multilayer Perceptron (MLP), Pruned decision trees (C4.5), Random Forests (RF), Logiboost (LB) and Bagging (BAG) with pruned C4.5 trees as base classifiers, k-Nearest Neighbor (k-NN) with one neighbor, and Support Vector Machine (SVM) are employed. **Results** RF yields the best classification results. The sensitivity, specificity, precision, recall, F1-measure of the RF are 92.3%, 92.3%, 92.2%, 92.3%, 92.2% respectively. The best classification latency, which is 0.01s, is achieved by NB, C4.5, BAG, and k-NN. In the deterministic approach, we have developed a set of predefined tree-based algorithms to rectify the shortcomings of classical machine learning algorithms. Experimental results indicate that the deterministic algorithm is able to classify direct and various models of indirect travel with 98.2% sensitivity, 98.1% specificity, 98.2% precision, 98.2% recall, 98.2% F1-measure, and 0.0003s classification latency. **Conclusion** The deterministic algorithm is simple to implement and highly suitable for real time applications aiming to monitor wandering behavior of PWD in long term care settings.

Keywords: wandering patterns, wandering behavior, dementia, classification

Martino-Saltzman[1] systematically evaluated the travel patterns of 40 nursing home residents with dementia, 24 of whom were identified by nursing staff as wanderers or suspected wanderers. The remaining 16 residents were non-wanderers. Travel was monitored continuously for 30 days and four basic travel patterns were observed: direct travel, lapping, random travel, and pacing. Travel efficiency (percentage of direct travel) was significantly related to cognitive status, with inefficient travel most prevalent in severely demented participants. The findings have facilitated further research in geriatrics and have been used in various dementia-related studies.

Travel patterns of PWD

First, inefficient travel (or wandering) patterns including lapping, pacing and random were operationally used to define wandering behavior of People with Dementia (PWD). Clinically, wandering is defined as a "syndrome of dementia-related locomotion behavior having a frequent, repetitive, temporally-disordered and/or spatially-disoriented nature that is manifested in lapping, random and/or pacing patterns"[2]. This 'comprehensive and integrated' definition was proposed in 2007 at the International Consortium for Research on Wandering on the basis of analysis and synthesis of terms and definitions from 183 published papers. The provisional consensus definition of dementia-related wandering was proposed to guide knowledge development across fields.

Second, wandering patterns combined with other spatial and temporal parameters can identify different types of wanderers and provide useful information on the wanderer's cognitive performance. In a longitudinal study, Algase et al.[3] videotaped ambulation episodes of 181 PWD and manually coded the wandering episodes (pacing, lapping, and random types). Parameters including rate, duration, and time of the day of wandering, drawn from the set of wandering episodes, were used to classify wanderers. Using principal component analysis, the authors distinguished three types of wanderers (classic, moderate, and subclinical). Cluster validation revealed differences between types of wanderers as well as non-wanderers in degree of cognitive impairment, mobility, and health indices[3].

Third, detailed findings from several medical studies[4-8] lent evidence to the correlation of wandering patterns and cognitive performance. Random-pattern wandering is considered the most serious symptom in PWD, followed by lapping and pacing. Increasing amounts of random-pattern wandering reflect higher degree of global (bilateral) hippocampal damage in humans and serve as a potential marker for dementia progression[4]. Frequent lapping signifies a deteriorating ability to way-finding in PWD and can be used as a predictor for unilateral hippocampal damage[5-6]. Pacing is not associated with levels of cognitive impairment, but is more indicative of agitation and anxiety[7-8]. Direct patterns are efficient but their character changes might also indicate cognitive deficits.

In the latest study[9], Lee et al. explored the relationship between observable emotional expression and wandering behavior of PWD. They found that positive emotional expression was positively related to wandering rates (wandering episodes per hour), whereas negative emotional expression and higher cognitive status were negatively related to wandering rates. Their results suggest that wandering behavior was related not only to cognition but also to emotional expression.

The need for an automated algorithm

The demand for an automated algorithm to classify travel patterns of PWD into direct and various modes of indirect travel arises from deficiencies of efficient methods to recognize these patterns.

In the clinical settings especially scientific environments, observational method is the most common (probably the only) approach to measure wandering behavior. Nursing scientists apply both direct and videotape observation techniques to study wandering of PWD. In both cases, vigilant observers or coders (primarily undergraduate nursing students) are hired to manually capture and document wandering episodes according to a predefined protocol. These coders practise in advance and are retrained until they match the researchers' coding standard. The protocol to capture wandering episodes is extracted as follows: "For every walking episode within each observation period, they (coders) identified and timed the start (i.e., three steps forward) and stop; i.e., after a period of walking momentum in any direction, there is (i) no stepping action or (ii) no forward momentum for 15s while participants steps in place. At the stop, coders then assigned a code for the pattern (random, lapping, pacing, direct). All (except direct) represented wandering"[10]. Wandering is then quantified by a rate parameter: number of wandering episodes per hour.

While observational method offers the greatest detail and information on aspects of wandering, it has several drawbacks. First of all, such method is suitable for institutional or research use and is restricted to studying mainly one patient at a time. Secondly, it requires time, labor, and money to hire and train coders. Thirdly, the tedious coding process is highly error-prone if the coder has to continuously observe the subject over a long period of time. Lastly, direct observation or cameras in people's homes in particular are considered extremely invasive and are viewed negatively. These drawbacks render this approach unsuitable for long term monitoring and management of wandering behavior.

In long-term care settings, wandering patterns are usually overlooked by care staff due to insufficient knowledge of wandering and unawareness of its negative effects on the subject's well-being. About 66% and 94% of PWD in developing and developed countries, respectively, are cared by informal caregivers, many of whom do not receive dementia or wandering-related education, training, and support services[11]. Moreover, there is a shortage of caregivers for PWD and they have to juggle multiple responsibilities and duties. Therefore, care staff generally cannot afford the time to surveillance wandering behavior of PWD[12]. In a survey of caregivers for PWD, wandering was listed as one of the top three most difficult behavioral problems to manage especially when this behavior occurs several times a day[13].

Previous studies have used ultra wideband radio frequency identification (UWB-RIFD) technology to measure two specific wandering behaviors: lingering near exit doorways and shadowing others who may exit[14,15]. In this paper, we automatically classify direct and various modes of indirect travel from room-to-room movement data collected by an active RFID system[16,17]. To the best of our knowledge, there was no such computerized program prior to our work. We believe the methodology is able to assist clinicians and care professionals in managing wandering behavior of PWD, without incurring the time and financial costs of observational method and the tedious pattern coding process associated with it.

Paper structure

This paper is structured as follows. First, we introduce the travel patterns constituting wandering in Martino-Saltzman (MS) model and describe the dataset as well as the ground truth used in the experiments. Second, we present a machine learning approach for classifying travel patterns based on eight classical algorithms. We also elaborate on the experiments and discuss the performance of all the employed algorithms.

Third, we present an improved deterministic predefined tree-based algorithm to classify travel patterns and its performance results. Finally, we conclude the paper and plan future works.

Background and dataset

MS model of wandering patterns

This section presents a typology of wandering. We will first define three concepts: location, movement, and episode. A 'location' may be represented in terms of 3-D space coordinates or broad regions (e.g. kitchen, dining room, etc.). A 'movement' is defined as moving from the present location to the next immediate location. Each 'travel episode', be it wandering or non-wandering, consists of one or more consecutive movements. Each episode has a start location and a stop location.

For convenience, we denote L_1, L_2, L_3, and L_4 as locations. Martino-Saltzman et al.[1] observed the spatial movements of both wanderers and non-wanderers with dementia and identified four patterns of independent travel in wandering subjects (*Figure 1*):

(i) Direct: a single straightforward path from one location to another without diversion. An episode comprising two or more consecutive direct paths to different locations is also considered as direct. A travel path that passes through the same location twice or more is not considered direct because one of the sub-trajectories is redundant. For example, a direct path from L_1 to L_4 includes passing through L_2 and L_3. Thus an episode involving $L_1L_2L_3L_4$ is direct. If an episode takes the path $L_1L_2L_3L_1L_2L_3L_4$, the first sub-path from L_1 to L_3 and back to L_1 is considered redundant and inefficient. Therefore, direct pattern would comprise single straightforward path or a path that moves through different locations. An episode with the path $L_1L_2L_1$ is direct.

(ii) Pacing: a repeated path back and forth between two locations. We specify that a pacing pattern would include more than two (at least three) consecutive to-and-fro movements. For example, the path $L_1L_2L_1L_2$ is classified as pacing since it has three repetitive movements: L_1 to L_2, L_2 back to L_1, and L_1 to L_2 again.

(iii) Lapping: a repeated circular path involving at least three locations[3]. A lapping pattern would contain at least two repeated circular routes involving at least three different locations, either in the same or opposite direction. For instance, the paths $L_1L_2L_3L_1L_4L_1L_2L_3L_1$(same direction) and $L_1L_2L_3L_4L_3L_2L_1$ (opposite direction) are considered as lapping;

(iv) Random: a continuous path with multiple locations in no particular order. A random pattern must not be a direct pattern and contain at least one location which is repeated at least twice.

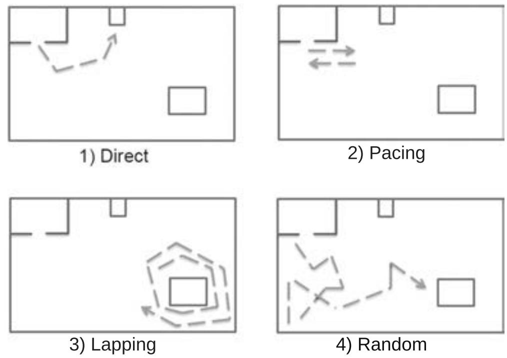


Figure 1. Travel patterns of nursing home wanderers; The plan view of a room is represented by a rectangle; Smaller rectangles represent different parts of the room and the dash lines show the travel paths

Due to these two conditions, lapping and pacing patterns are subsets of random patterns.

With respect to distance travelled and time taken, direct pattern is efficient travel and is not regarded as wandering[1]. The other three patterns (random, lapping, and pacing) are inefficient and they constitute different types of wandering. Martino-Saltzman framework has been widely used by many experts[1-7] in the field of gerontology and nursing research to study wandering behavior of PWD. It has influenced further research and is one of the very few empirical typologies to measure and quantify wandering[4]. Subsequent studies[5-8] have demonstrated that affected wanderers often exhibit more than one inefficient pattern.

Dataset

Subjects' characteristics

Movement datasets of five nursing home residents with different stages of dementia are used in our study. This includes 2 males and 3 females of 72.4±8.4 years of age. The Mini-Mental State Examination showed signs of dementia in all five subjects (2 were diagnosed as having vascular dementia, the other 3 Alzheimer's disease)[16,17]. All could walk independently[16].

Subjects' movement data

The movement data were collected by Makimoto et al.[16,17]. Active RFID activity monitoring systems (Power Tag, Matrix Co. 6-1-2 Nishi Tenma, Kita-ku, Osaka, 530-0047 Japan) were installed at two dementia care units in Japan and Korea. Antennas were set up on the ceilings of all the rooms in the units. Individual RFID tags (measuring 2.8 cm x 4.2 cm x 0.68 cm) were worn by the subjects in the back collar of their shirts. A personal computer system then recorded the movement information including the tag ID, the tag receiver ID, time and date. Each tag receiver

Table 1. Specifications of the Power Tag RFID system[31]

Tag receiver	
Receiving frequency (RF)	304.2, 309.9, 314.26 MHz
Number of RF inputs	1
Type of modulation	Frequency shift keying
Signal input sensitivity	-90dBm to -30dBm
Data output	RS232C 9600bps
Power supply	AC 90/110V 50/60Hz
Trigger generator	
Output	1 Arms/93.75kHz
Power supply	DC 12V/2A AC Adapter
RF tag	
Output frequency	304.2MHz or 309.9MHz or 314.26MHz
Modulation	FSK 60kHzp-p
RF output level	500i V/m at 3m offset (Weak Radio Wave)
Wake up signal	>100m Vp-p 93.75kHz by electromagnetic
Signal rate	93.75kps
Power supply	DC3.3V ML1220 rechargeable battery

ID (or location ID) uniquely identified the room visited by the monitored subject.

The Power Tag system (*Table 1*) is able to monitor the whereabouts of a subject and the rhythm of daily activities such as walking distance per day and frequency of toileting. Makimoto et al.[16,17] employed the system to record the room-to-room movements of monitored subjects as they moved within the care units over a 24-hour period.

The graphical representation of the movements for each subject (labelled as A, B, C, D, and E) is reproduced in *Appendix A*. Subjects A, B and E were monitored in the same unit in Japan whereas subjects C and D were monitored in another same unit in Korea.

We label the semantic locations of the two units by integer numbers for computational purposes (*Table 2*).

Table 2. Numerical labels of semantic locations in care units in Japan and Korea

Japan		**Korea**	
Location	**Label**	**Location**	**Label**
Room 303	1	Room 5304	1
Toilet	2	Toilet 5304	2
Function hall	3	Activity Room 2	3
Dining room	4	Corridor 5301	4
Room 302	5	Corridor activity room 2	5
Bathroom	6	Nursing station	6
Room 301	7	Corridor 5306	7
Room 305	8	Room 5306	8
Emergency exit (ee)	9	Corridor 5309	9
		Activity room 1	10
		Corridor 5303	11
		Elevator	12
		Corridor undefined	13

Ground truth estimation

In total, 120 hours (24hrs per subject x 5 subjects) of data were recorded by Makimoto et al.[16] This original movement dataset consists of 1163 instances. Each instance has four attributes: the time, date, tag ID and location ID.

Two steps are needed to establish the ground truth. The first step is to extract every travel episode bounded by its start and stop locations. This is done using an automatic episode segmentation algorithm. The second step is to apply MS model manually to map each episode to one of the four patterns: direct, random, lapping, or pacing.

Step 1: Automatic episode segmentation (temporal information was used in this step). To segment an episode, we need to identify its start location and stop location. Ideally, an accelerometer attached to the monitored subject can be used to accurately determine the start/stop locations of each episode. However, even though such motion information is not available in the movement records used in this study, we are able to obtain good estimation of the start/stop locations from the temporal information.

For each subject's dataset, we assumed the location of the first data instance was the start location of the first episode. The start location of any subsequent episode would be the immediate location after the stop location of the previous episode. The stop location of an episode was defined as one where the subject spent more than 15s (same as the value used in standard protocol by gerontologists[10]).

Suppose the RIFD system recorded that the subject entered location L_1 at time t_1 and entered the next immediate location L_2 (after being at L_1) at time t_2 ($L_1 \# L_2$ and $t_1 < t_2$). The total travel time from L_1 to L_2 ($t_2 – t_1$) is bounded by:

$$t2\text{-}t1 \leq totalTh = 15s + maximumDirectTravelTime + wanderingOffsetTime \quad [1]$$

With:
15s = stop threshold; maximumDirectTravelTime = the maximum time to travel directly from L_1 to L_2; wanderingOffsetTime = the time the subject may wander on the direct path between two furthest locations. It is zero if there is no wandering.

Therefore, whether or not L_1 is the stop location of an episode can be easily determined by comparing the total travel time (t_2-t_1) with totalTh.

In indoor environments such as dementia care units[16,17], the maximumDirectTravelTime is bounded by the ratio of the maximum distance between neighboring locations (rooms) and the minimum walking speed of the subject. The walking speed of a subject can be retrieved from the RFID system. For the current dataset, we used the empirically obtained threshold: totalTh=55s (maximum distance=13m, minimum walking speed=0.53m/s, and wanderingOffsetTime=15s) to do automatic episode segmentation.

We apply the technique described above to all five subjects' datasets. In total, 220 travel episodes were identified (40±30 episodes per subject). Each travel episode consists of a sequence of locations the subject had travelled.

Step 2: Manual classification of an episode's travel pattern (spatial information was used in this step). Based on the sequences of locations travelled within an episode, we manually classified the travel patterns of each episode into one of four types: direct, random, lapping, and pacing. We observed the sequence of travelled locations and the contextual information (e.g. meal times, layout of the care unit, the intended destination the subject wanted to travel to) to distinguish between direct and inefficient travels. If it was an inefficient travel, we further checked for pacing, lapping, or random pattern. The pattern of the episode was then concluded accordingly (*Table 3*). If there were multiple patterns within an episode, the concluding pattern was the one which had the most number or the most severe one (ascending order of patterns' severity: pacing, lapping, and random). These manually classified results serve as ground truths for evaluation of the automated algorithms.

Machine learning approach

The machine learning approach for classifying travel patterns consists of two stages: (i) feature extraction, and (ii) classification.

Feature extraction

From the location sequence, we compute four representative features for each travel episode. The features are entropy (F1), repeated locations (F2), repeated directions (F3), opposite directions (F4).

The first feature, F1 measures the entropy of each episode. Entropy is the average information or unpredictability in a random variable. Therefore, entropy can be used to represent the randomness of movement in an episode.

The second feature, F2 aims to distinguish direct pattern from other types of patterns by counting the number of repeated locations in an episode. This is based on the fact that if a person keeps revisiting a number of locations in an episode of continuous movements, that episode is considered inefficient travel. Pacing and lapping patterns are repetitive movements in back-and-forth and circular manner. For these two types of patterns, not only are the locations of travel repeated but also the directions of movement (or travel). Therefore, we use two features, F3 and F4, to represent the repetitiveness of travel di-

Table 3. Ground truth estimation of the 4 travel patterns of the 5 subjects

Statistics	Subject					
	A	B	C	D	E	All
			Direct			
Number of episodes	29	22	40	13	22	126
Range, s	14–127	36–141	13–89	12–31	9-84	9–141
Mean±SD	39±37	45±49	28±27	18±8	23±26	32±34
			Pacing			
Number of episodes	2	4	10	1	0	17
Range, s	135-156	61–326	26–115	66	0	26–326
Mean±SD	146±15	169±112	70±24			102±69
			Lapping			
Number of episodes	1	5	34	1	5	46
Range, s	386	182–781	70–1545	105	102–234	70–1545
Mean±SD		423±223	351±309		179±53	335±282
			Random			
Number of episodes	0	5	13	7	6	31
Range, s	0	256–674	87–173	70-220	130–430	70–674
Mean±SD		383±168	126±27	103±52	239±124	184±132
			Total			
Number of episodes	32	36	97	22	33	220
Range, s	14–386	36–781	13–1545	12–220	9–430	9-1545
Mean±SD	56±75	158±191	159±233	51±50	86±108	122±186

rections in each episode. Travel direction is the directional vector of two consecutive locations.

F3 counts the number of repeated travel directions in each episode. F4 counts the number of pairs of opposite travel directions. For example, in an episode, travel directions of two movements - one from location L_1 to location L_2 and another from location L_2 back to location L_1 - are considered a pair of opposite travel direction. Feature F4 is needed because the subject can pace and lap in opposite directions.

Now, we provide details on how the features are measured. Suppose that an episode with n-location sequence in a chronological order is represented as a vector

$$L = (L_1, L_2, \ldots, L_n), \quad [2]$$

whereas $L_i \neq L_{i+1}, i = \overline{1, n-1}$

and L_i are labels from Table 2.

From the vector L, we obtain:

-The direction vector

$$D=((L_1,L_2),(L_2,L_3),\ldots,(L_{n-1},L_n)) \quad [3]$$

-The set of distinct elements in vector L,

$$S_L=\{L_i, 1\leq i\leq n \mid L_i \in L\} \quad [4]$$

-The set of distinct elements in vector D,

$$S_D=\{(L_i,L_{i+1}),1\leq i\leq n\text{-}1 \mid (L_i,L_{i+1}) \subseteq D\} \quad [5]$$

-The frequency of occurrence of each element in S_L, $f_i = (\textit{number of occurrences of } L_i \textit{ in } L)/n$, $1\leq i\leq n$ [6]

Then, the four features are calculated as follows.

-Feature F1 is the entropy of the episode:

$$F1 = -\sum_{i=1}^{n} f_i \log f_i \quad [7]$$

-Feature F2 counts the total number of locations visited which are repeated. This is equal to the total number of elements n minus the number of unique elements in L (i.e. the cardinality of S_L).

$$F2 = n - \|S_L\| \quad [8]$$

-Similar to F2, F3 is calculated as:

$$F3 = n - 1 - \|S_D\| \quad [9]$$

-F4 is meant to distinguish travel in the same or opposite directions.

$$F4 = \|\{1\leq i\leq n\text{-}1 \mid \exists\, j, 1\leq i<j\leq n\text{-}1 \wedge L_i=L_{j+1} \wedge L(i+1)=Lj\}\| \quad [10]$$

Feature selection and classification

Experiments and evaluation

We have experimented with different combinations of the four features, and found that the best results were achieved by using all four features for classification. We tested eight classical machine learning algorithms including NB, MLP, C4.5, RF, LB, BAG, k-NN, and SVM. All algorithms were tested using Weka freeware[18]. The best performing configurations of all the eight classifiers are also reported.

NB classifier is based on Bayes's theorem with the assumption that the effect of a particular feature on a given class is independent of other features. It is made to simplify the computation involved and, in this sense, is called 'naïve'. Given a sample X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is X is predicted to belong to the class C_i if and only if it is the class that maximizes $P(X|C_i)P(C_i)$. $P(C_i)$ is estimated by counting the frequency with which each class C_i occurs in the training data. Based on the assumption that features are independent given the class, $P(X|C_i)$ is the product of the probabilities for the individual features given class C_i[19].

MLP is a neural network that mimics a neuronal organizational structure. It uses back propagation to train the network aiming to minimize the squared error between the network output and target values[20]. The network configuration used in our experiments is learning rate=0.3, momentum=0.2, epochs=500.

Decision trees represent the set of classification rules in the form of a tree. It uses information gain to measure how well a given feature separates the training examples into their target class[21]. We used the J48 Decision trees provided with the Weka software to classify travel patterns. The configuration used in our experiments is confidence factor=0.25, the minimum number of instances per leaf=2.

RF is an ensemble classifier that consists of many decision trees and outputs the most popular class. A tree is grown from independent random vectors using a training set, resulting in a classifier. After a large number of trees is generated, random forest outputs the class that is the mode of the class's output by individual trees[22,23]. The configuration used in our experiments is the number of trees to be generated=10, the random number seed=1, the number of attributes to be used in random selection=0.

LB[24] and BAG[25] are ensemble methods that include multiple 'base' classifiers, each of which covers the complete input space. Each based classifier is trained on a slightly different training set and the predictions or classifications of all classifiers are then aggregated to produce the single output. The simplest way to aggregate is to take a vote (e.g. a weighted vote). BAG and LB both adopt this approach but they derive the individual classifiers in different ways. In BAG, the classifiers receive equal weight, whereas in LB weighting is used to give more influence to the more successful classifiers. The base classifier used in experiments with BAG is the fast decision tree learner, and the model trees in experiments with LB used M5Rules.

k-NN calculates instances using Euclidean distance and corresponds to points in an n-dimensional space. The algorithm assigns a class label to a data point that represents the most common value among the k training examples nearest to the data point[22]. We used k=1 in our experiment.

SVM maximizes the margin between the training examples and the class boundary. SVM generates a hyperplane which provides a class label for each data point described by a set of feature values[26]. The kernel used in our experiments is the normalized poly-kernel.

The 10-fold stratified cross validation technique is applied and results are obtained in term of five validation metrics: precision, recall, specificity, F1-measure, and latency. The classification performance is based on individual episodes of movements, i.e. for each episode the classifier output is compared to the reference annotation (the ground truth).
For each type (class) of travel patterns, let's call episodes of its class (in the ground truth) 'positives' and episodes not belonging to this class 'negatives'.

Then, 'positive' episodes that are correctly (incorrectly) labelled by the machine learning algorithms are counted as True Positive (False Negative) (shortened as TP and FN) whereas 'negative' episodes that are classified by the algorithms as 'negative' episodes ('positive' episodes) are counted as True Negative (False Positive) (shorten as TN and FP).

We measure:
(i) the precision (or Positive predictive value)

$$Prec=TP/(TP+FP)) \qquad [11]$$

which represents the proportion of 'positive' classified episodes that are relevant;
(ii) the recall (True Positive Rate or sensitivity)

$$Reca=TP/(TP+FN) \qquad [12]$$

which measures how good the classifier is at detecting 'positive' episodes;
(iii) the specificity (or True Negative Rate)

$$Spec=TN/(TN+FP) \qquad [13]$$

which evaluates how good the classifier is at avoiding false alarms.

F1-measure

$$F1 = 2\cdot Prec\cdot Reca/(Prec+Reca) \qquad [14]$$

takes into account the precision and recall rate for each class.

In addition, we evaluate the algorithms in terms of classification latency, which is the time delay between the start of a travel episode in the reference dataset and the start of a detected (classified) wandering episode by the application.

To report the overall performance metrics, we take the weighted average (or weighted arithmetic mean) of these measures from all classes, i.e. weighting the measure of each class of patterns by the proportion of instances there are in that class. This is done to avoid the problem of inflating the accuracy or recall values of classes with high recall values but few instances.

Feature selection
Better results are obtained by using all four features (*Table 4*). Thus features F1, F2, F3 and F4 are used in the subsequent experiments.

We also tried using common features including mean, standard deviation, and variance to classify travel patterns. Not surprisingly, these features do not represent the data well because the location labels do not capture the spatial relations of the physical locations.

Classification results
We use confusion matrices of the eight classifiers to provide a more detailed picture of the errors

Table 4. Evaluation of the classification algorithms in Experiment (Exp)1 , using the features F2-4 (repeated locations, repeated directions and opposite directions), and in Exp 2, using all 4 features (F1=entropy, added); NB= Naïve Bayes.; MLP= Multilayer Perceptron; C4.5= Pruned decision trees; RF= Random Forests; BAG=Bagging; LB=Logiboost ; k-NN=k-Nearest Neighbor; SVM=Support Vector Machine

Criterion	Exp	Classifier							
		NB	**MLP**	**C4.5**	**RF**	**BAG**	**LB**	**k-NN**	**SVM**
Sensitivity / recall	1	77.7	84.5	90.0	91.4	89.5	91.4	91.8	84.1
	2	80.0	86.8	91.4	92.3	91.4	91.4	88.6	88.6
Latency, s	1	0.01	0.24	0.01	0.02	0.01	0.03	0.01	0.04
	2	0.01	0.33	0.01	0.03	0.01	0.05	0.01	0.05
Precision	1	76.2	82.9	90.7	91.3	90.9	91.2	91.7	80.8
	2	80.4	86.3	91.2	92.2	91.3	91.2	88.3	89.3
Specificity	1	77.7	84.5	90.0	91.4	89.5	91.4	91.8	84.1
	2	80.0	86.8	91.4	92.3	91.4	91.4	88.6	88.6
F1 measure	1	76.4	83.3	90.2	91.3	89.9	91.2	91.7	81.7
	2	87.8	85.6	91.2	92.2	91.4	91.3	88.1	88.6

Table 5. Confusion matrices of travel patterns for the different classifiers: BAG (Bagging), C4.5 (Pruned decision trees), k-NN (k-Nearest Neighbor), LB (Logiboost), MLP (Multilayer Perception), NB (Naïve Bayes), RF (Random Forests), and SVM (Support Vector Machine); D=direct; P=pacing; L=lapping; R=random

BAG

Ground truth	Classified as D	P	L	R
D	124	0	0	2
P	1	12	2	2
L	0	2	41	3
R	2	0	5	24

C4.5

Ground truth	Classified as D	P	L	R
D	126	0	0	0
P	1	12	2	2
L	0	3	40	3
R	2	2	4	23

k-NN

Ground truth	Classified as D	P	L	R
D	125	0	0	1
P	3	12	1	1
L	3	2	39	2
R	8	0	4	19

LB

Ground truth	Classified as D	P	L	R
D	125	0	0	1
P	1	12	3	1
L	0	3	39	4
R	2	0	4	25

MLP

Ground truth	Classified as D	P	L	R
D	125	0	0	1
P	1	12	2	2
L	0	1	43	2
R	4	9	7	11

NB

Ground truth	Classified as D	P	L	R
D	119	0	0	7
P	8	7	1	1
L	2	1	36	7
R	16	0	1	14

RF

Ground truth	Classified as D	P	L	R
D	125	0	0	1
P	1	14	2	0
L	0	2	41	3
R	2	0	6	23

SVM

Ground truth	Classified as D	P	L	R
D	120	0	6	0
P	1	12	3	1
L	1	1	41	3
R	3	0	6	22

Table 6. Results of 'Leave one subject out' experiment

Subject left out	Episodes of wandering #	Sensitivity	Specificity	Precision	Latency
A	32	1	1	1	0.01
B	36	0.833	0.937	0.862	0.01
C	97	0.897	0.974	0.924	0.01
D	22	0.727	0.606	0.813	0.01
E	33	0.909	0.929	0.928	0.01
Weighted average	220	0.873	0.889	0.905	0.01

generated by the classification process (*Table 5*). The RF algorithm yields the best classification sensitivity or accuracy (92.3%). The RF algorithm indicated slightly higher accuracy compared to the decision trees algorithm. However, the decision trees algorithm is much simpler and transparent in decision making process compared to the RF. The RF algorithm introduces a much higher complexity in the decision making process, since the result is obtained using 10 decision trees and a mode voting procedure (i.e. the classified class is the mode of the classes output by individual trees).

The RF classification algorithm is also evaluated using the leave one subject out method. In each leave one subject out experiment, dataset of a subject is used for testing and the combined datasets of other four subjects are used for training. The sensitivity, specificity, precision, F-1 measures per subject, for each leave one out experiment, along with the average statistics are presented in Table 6. The weighted average accuracy for the leave one subject out experiment is 87.3%.

Rationale

First of all we explain why classifiers based on decision trees produce high accuracy in classifying travel patterns. Figure 2 reproduces the pruned tree generated by the decision trees algorithm. The leaves represent class labels or travel patterns, and branches represent conjunctions of features that lead to those class labels.

Figure 3 represents the number of repeated locations of the entire 220 episodes. It clearly shows that direct episodes are distinguished with other episodes because the amplitudes of most direct episodes are zero. Direct patterns are efficient travel; therefore, it is highly likely that there is no repeated location in the sequence of locations of direct episodes. Hence, using F2, we can easily separate episodes into two main groups: direct (marked with long dashed rounded rectangle) and non-direct. The main task now is to classify pacing, lapping, and random from non-direct episodes.

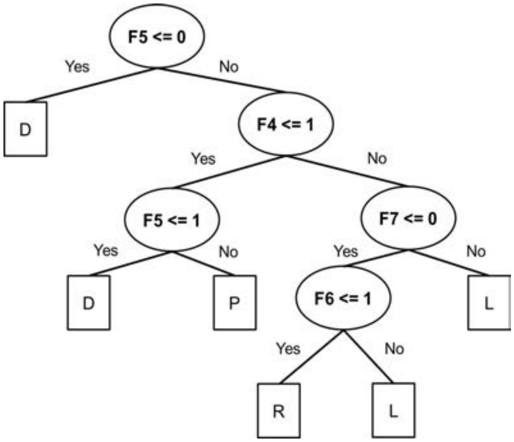


Figure 2. Pruned tree created by C4.5 algorithm

Using feature F1, we can discern pacing from lapping and random. Pacing is repetition movements between two locations. Therefore, the occurrence frequency of each location should be equal. Thus, the entropy of pacing episodes should be constant or equal to one. This is obviously shown in Figure 3b in which entropy of pacing episodes is one (marked with long dashed rounded rectangle) and entropy of lapping and random episodes is more than one.

The main task left now is to distinguish lapping and random from each other. Features F4 and F3 are mainly used for this task. In Figure 3c, almost half of lapping episodes are separated from random episodes. They are episodes whose amplitudes of F4 are above 2. These episodes are highlighted in the long dashed rounded rectangle. Other lapping episodes, whose F4 values have the same amplitude range with the ones of random episodes, are highlighted in the rounded rectangle (*Figure 3c*). We have to discern these lapping episodes from random episodes. Respectively, these lapping episodes are also indicated using the rounded rectangle in Figure 3d. From this figure we can see that the amplitudes of feature F3 of these lapping episodes are more than one whereas the ones of random episodes are almost zero or below one. Hence, we can classify these lapping and random episodes into their corresponding classes. The classified random episodes are marked with long dashed rounded rectangle in Figure 3d.

Misclassifications and errors

Misclassifications are caused by three shortcomings of the employed algorithms.

First, they are not able to handle several multi-pattern episodes. This occurred 29.4%, 31.6% and 32.0% respectively for the C4.5, RF and 'Leave one subject out' experiments. Examples of multi-pattern episodes are (dining, hall, dining, hall, toilet, hall, toilet) and (302, 301, bath, 301, 302, 303, hall, dining, hall, 303, 302). The first episode comprises two sub-patterns: pacing between the dining room and function hall, and pacing between the hall and the toilet. The second episode comprises two sub-patterns: lapping in the opposite direction between rooms 302, 301 and bath, and lapping in the opposite direction between rooms 302, 303, hall, and dining. In these two multi-pattern episodes, some locations ('hall' in the first episode, '302' and '303' in the second episode) belong to both patterns, which the decision trees based algorithms do not recognize. Therefore, these algorithms misclassified the first episode as 'lapping' and the second episode as 'random'. Another example is (ee, 303, 302, 303, ee, toilet, hall, ee, 303, 302, 301, 302, 303). 2 lapping sub-patterns (ee, 303, 302, 303, ee) and (303, 302, 301, 302, 303) are separated by a direct sub-pattern (toilet, hall, ee). However, the algorithms did not recognize these and misclassified as random.

Second, the algorithms are not adaptive to learn/classify episodes with few or almost zero training instances. This was the case in 35.3%, 52.6% and 24.0% respectively in the three experiments. Examples of misclassified episodes caused by the shortcoming 2 are (dining, hall, 303, hall, dining, hall, 303) and (C5306, AR2, NS, R5304, NS, CAR2, C5301, C5309, C5306, C5303). The employed algorithms are not able to recognize the first episode as lapping in the same direction but misclassified it as pacing. We hypothesize that the algorithms recognized the repeated subsequences (dining, hall), and (hall, 303) in the first episode. However, they did not realize that these subsequences need to occur continuously in order to constitute a pacing pattern. Therefore, they misclassified it as pacing. In the second example, the algorithms misclassified the random episode as lapping. Obviously, the second episode cannot be classified as lapping because there is no loop in the episode.

To rectify the first two shortcomings, we need to have an adaptive algorithm that is able to separate the individual sub-patterns, classify these sub-patterns accordingly and then aggregate them together so as to make the correct conclusion.

Third, the algorithms do not incorporate contextual information (e.g. layout of the monitored area) to reason about the efficiency of travel episodes. The three experiments showed that this occurred in 35.3%, 15.8% and 44.0% respectively. An example episode related to the shortcoming 3 is (R5304, NS, CAR2, C5309, AR2, C5303). The employed algorithms classify it as

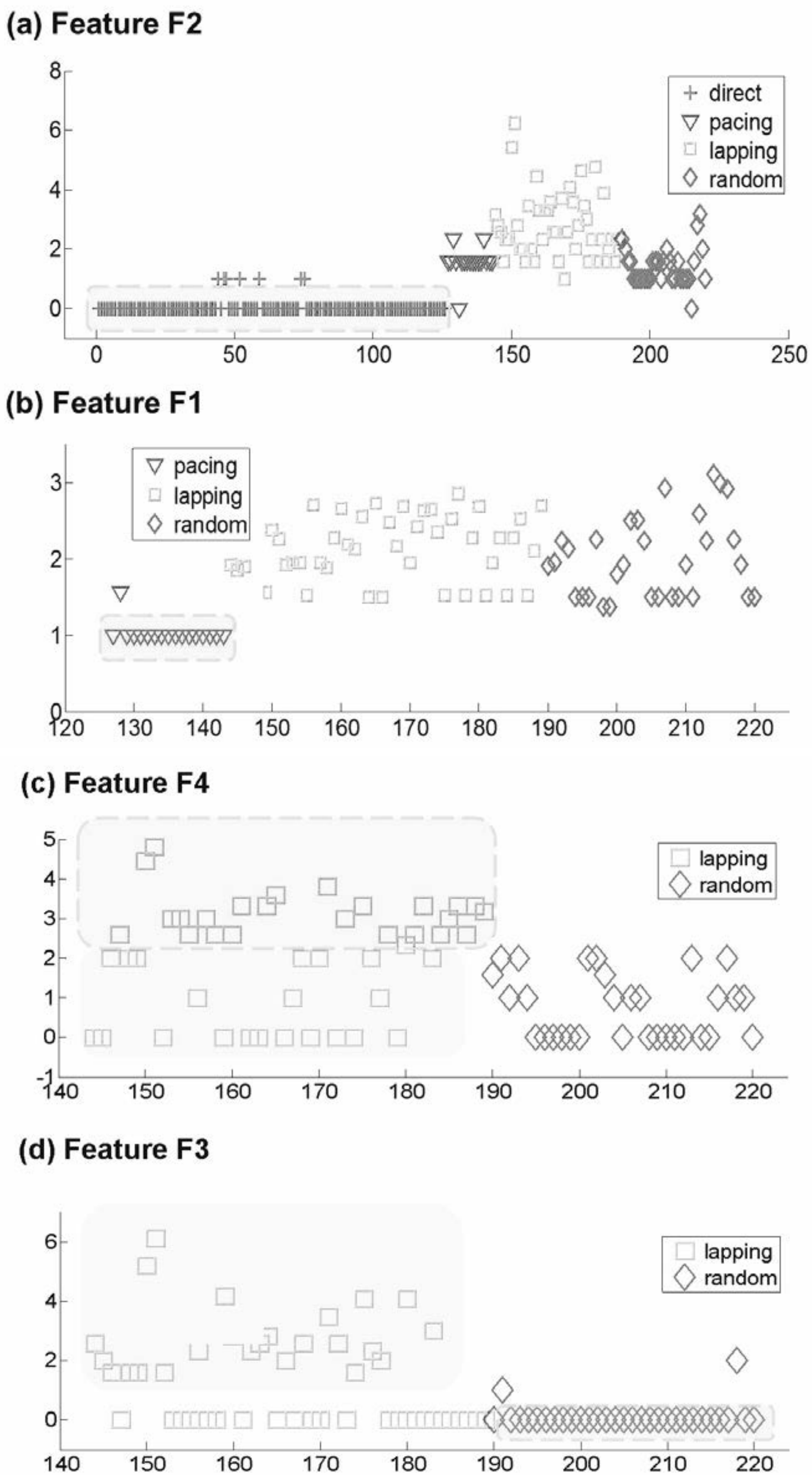


Figure 3. Features F1-F4 used in classifiers; horizontal axes represent the travel episode with labelled patterns; vertical axes denote: F1: raw entropy value, F2-F4: log (value of the amplitude +1) of each feature (log scales are used for easy visualization)

direct, which is not correct because it is indeed possible to reach the destination corridor C5303 from the room R5304 without going through locations NS and C5309. Therefore, it is an inefficient episode and should be classified as random.

There are two approaches to detect such inefficiency. The first and direct approach is to base on the layout of the monitored area. If the layout shows there is a direct or more efficient path between C5303 and R5304, we can apply shortest path algorithms to find the direct path and make the conclusion.

The second approach is applied when there is no layout or map available. In this case, we can base on the historical movements of the subjects to reason if there is a direct path between C5303 and R5304 or any other more efficient path, for instance (R5304, CAR2, AR2, C5303). This solution requires retrieving and searching previous episodes to detect such direct or more efficient paths. In this paper, we use the second approach to reason about this case because there is no layout available for the subjects monitored in the dementia care centre in Korea. Nonetheless, the second approach does not work for all the cases especially when there is no historical movement of the path needed to be searched or retrieved.

Deterministic predefined tree-based algorithm

Algorithm's design

The goal of algorithm's design is to rectify the above shortcomings so as to improve the performance of decision trees algorithm.

Single- and multi-pattern episodes

There are two cases: single-pattern episodes and multi-pattern episodes. A single-pattern episode has only one pattern from start location to end location. For example, episodes $L_1L_2L_3L_4$or $L_1L_2L_3L_4L_3L_2$ are single-pattern because there is only one pattern appearing in the entire episode (direct for $L_1L_2L_3L_4$and lapping for $L_1L_2L_3L_4L_3L_2$). An episode such as $L_1L_2L_3L_4L_3L_2L_3L_2$ is multi-pattern because there are two patterns, lapping ($L_2L_3L_4L_3L_2$) and pacing ($L_3L_2L_3L_2$) in the episode.

Since there is only one concluding pattern (direct, pacing, lapping, or random) in single-pattern episodes, we propose to design a sequential algorithm to do the classification. Technically, we assume that the sequential algorithm consists of four individual modules. Each module is responsible for classifying one type of pattern. By sequentially applying these modules, we are then able to classify single-pattern episode into the corresponding concluding pattern.

We consider each multi-pattern episode as a concatenation of single-pattern episodes. We can therefore apply the deterministic algorithm to classify the individual single-patterns or sub-patterns within a multi-pattern episode and aggregate these sub-patterns together to result in the concluding pattern.

To accomplish the proposed deterministic algorithm, there are three components needed to be addressed. First, we need to design modules or algorithms to check for direct, pacing, lapping, and random patterns. Second, we need to determine the sequence or the order of applying these modules to classify both single-pattern and multi-pattern episodes. Third, we have to propose an aggregation scheme for classifying multi-pattern episodes.

The objectives of the first and third components (individual modules and aggregation scheme) are quite clear and the details will be explained in next sections.

In this paragraph, we illustrate why it is important to address the second component. If we imagine each module to check for each type of travel patterns is a leave in a tree, then determining the sequence of applying checking modules is indeed analogous to determining the pruned tree in the decision trees algorithm. Different sequences might then produce different classification results and have different impacts. To illustrate this, we use episode $L_1L_2L_3L_4L_3L_2L_3L_2$,

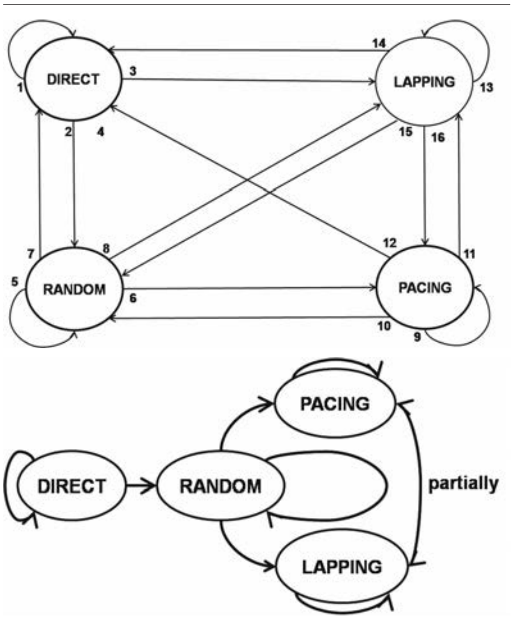


Figure 4. Transitional state diagram; (a) Complete version; (b) Demonstrating the transformation of travel patterns

which could be classified as direct ($L_1L_2L_3L_4$) and pacing ($L_3L_2L_3L_2$), or lapping ($L_2L_3L_4L_3L_2$) and pacing ($L_3L_2L_3L_2$). If the sequence of applying checking modules is lapping first followed by direct and pacing, we will get two sub-patterns direct and pacing, which is incorrect because they do not recognize that this is a composite episode with common locations L_3 and L_2. In other words, they do not rectify the first shortcoming. However, if the sequence of checking is direct first followed by lapping and pacing, it will yield correct results (lapping and pacing) and rectify the first shortcoming.

Theoretically, one can try all the enumerations of possible sequences but it is inefficient. We propose an effective approach to reason and identify the correct sequence of applying the checking modules. In the subsequent section, we first present the transformations of travel patterns. It describes how travel patterns evolve over time and space. Next we identify the sequence of applying checking modules. After that, we present the sequential algorithm in which the first and third components are detailed.

Transformations of travel patterns

There are four types of patterns or states. We have established the complete directed cyclic graph with four states (16 edges) to represent how patterns transform from one type to another type. The complete transitional state diagram has all 16 edges numbered (*Figure 4a*). The criteria to prune out edges which are impossible to exist are as follows.

A travel episode begins from a single start location. If the subject remains at the start location, it is a direct pattern itself. As the subject moves to new locations, the travel pattern could remain as direct or change to inefficient patterns. This explains why the edges 1, 2, 3, and 4 are retained. However, inefficient patterns cannot change back to direct pattern because they contain repeated locations, directions, or diversion. Hence, an episode always starts as direct and can evolve to inefficient travel but not the other way round. Therefore, the edges 7, 12, and 14 are removed.

By definition, pacing is a repeated path back and forth between two locations whereas lapping is a repeated circular path involving at least three locations. Thus, at least two consecutive locations must be repeated at least twice in order to constitute a pacing and/or lapping pattern. A path containing a sub-path of two consecutive locations that is repeated twice is already a random pattern (regardless of whether or not there is any sub-path or location between the repeated sub-paths). Such a path would remain as random or transit into pacing or lapping. Therefore, an inefficient pattern must be in the random state before it evolves into pacing or lapping. Due to this, we conclude that direct state cannot change directly to pacing and lapping states. In addition, pacing and lapping patterns are subsets of random patterns. Hence, the edges 3, 4, 10, 15 are pruned out. And the edges 2, 5 are retained.

Subsequently, we explain why the edges 9, 11, 13, and 16 are retained. By definition, pacing and lapping contain repeated continuous paths between two or more locations. In a lapping/pacing episode, if such paths are repeated the pattern is unchanged. This explains why the edges 9 and 13 are retained.

If a new path is visited during the episode, a new pattern, not necessary pacing or lapping, may be formed. In such cases, it would constitute a multi-pattern episode. If the new pattern is lapping or pacing, there is a partial partnership between the old lapping/pacing pattern with the new lapping/pacing pattern.

The partnership is illustrated using two examples: $L_1L_2\underline{L_1L_2}L_3L_4L_1L_2L_3L_4$ and $L_1L_2L_3L_4\underline{L_3L_2}L_3L_2$. For the former, the sub-path $\underline{L_1L_2}$ is part of a pacing pattern ($L_1L_2\underline{L_1L_2}$) and a lapping pattern ($\underline{L_1L_2}L_3L_4L_1L_2L_3L_4$). This showcases how a pacing state can partially transform to a lapping state. For the latter, the sub-path $\underline{L_3L_2}$ is part of a pacing pattern ($\underline{L_3L_2}L_3L_2$) and a lapping pattern ($L_2L_3L_4\underline{L_3L_2}$). In this example, a lapping state partially transforms to a pacing state.

The partnership also occurs between the old lapping (pacing) pattern and the new lapping (pacing) pattern. Examples are $L_1L_2L_3L_1L_2L_3$ $L_1L_2L_3L_1\underline{L_2L_3}L_4L_2L_3L_4L_2L_3L_4$ or $L_1L_2L_1L_2L_1\underline{L_2}L_3L_2L_3L_2L_3$. Hence, the edges 11 and 16 are retained.

We then obtain the pruned state diagram (*Figure 4b*). The state diagram depicts the transformations from efficient to inefficient travels in an ambulation episode as the wandering subject

Table 7. Six steps in state transformations of the ambulation episode $L_1L_2L_3L_4L_3L_2$

Step	Movement sequence	State	Pattern explanation
1	L_1	direct	Direct by definition
2	L_1L_2	direct	Direct by definition
3	$L_1L_2L_3$	direct	Direct by definition
4	$L_1L_2L_3L_4$	direct	Direct by definition
5	$L_1L_2L_3L_4L_3$	random	Random, due to the repeated location L_3
6	$L_1L_2L_3L_4L_3L_2$	lapping	Lapping, due to the circular path $L_2L_3L_4L_3L_2$.

moves from one location to another. The first movement from the start location initializes the state to direct. Depending on subsequent movements, the state may remain direct throughout the entire episode or change to random, then possibly followed by pacing or lapping. The state of the ambulation episode is continually updated until it reaches the end location. The final state label (e.g. random) is the episode's classification. For multi-pattern episodes, when a pattern is in pacing state, it either remains there or evolves to lapping and vice versa (*Figure 4b*). Table 7 illustrates the step-by-step process of classifying the path $L_1L_2L_3L_4L_3L_2$ as a lapping episode.

Algorithm's formulation

The deterministic tree-based algorithm

From the state diagram (*Figure 5b*) and the example in Table 7, we identify that we only need to have three checking modules for the deterministic algorithm. These modules are to check for direct, pacing, and lapping patterns respectively. If an episode does not belong to any of the three patterns, it should be random. Based on this observation and the pruned state diagram, we formulate the sequence of applying checking modules as follows. The algorithm will first check if the episode is direct. If not, it is an inefficient travel pattern and the algorithm will check if the episode is pacing or lapping respectively. The ambulation episode is concluded as random if it is neither pacing nor lapping. In fact, this sequence is highly similar to the pruned tree (*Figure 2*), where direct pattern is examined in the zero level, followed by pacing and lapping in the second level, then random in the third level. In other words, the state diagram is agreeable with the empirical result produced by decision trees algorithm. Due to this result, we name the proposed algorithm as the deterministic tree-based algorithm.

A multi-pattern episode is considered as a concatenation of single-pattern episodes. To classify multi-pattern episodes, we also start by checking if the entire episode is direct. If not, we check for single-pattern episodes that are pacing. This is done by checking for the longest repeated pacing paths, e.g. '$L_1L_2L_1L_2$' or '$L_1L_2L_1L_2L_1L_2$'. Then, we will check for single-pattern episodes which are lapping. This is accomplished by checking for the longest circular paths (in both same and opposite directions). We will mark all the longest pacing and lapping paths as checked. For the remaining unchecked paths in the multi-pattern episode, we will label them as random if there are repeated locations and directions in those unchecked paths (by definition).

The aggregation scheme for multi-pattern episodes is designed as follows. We count the number of occurrence of each type of inefficient patterns for the entire multi-pattern episode. The concluding pattern is the one with the highest number of occurrence. If more than one inefficient patterns have the highest number of occurrence, the conclusion is drawn based on the severity of inefficient patterns (random, followed by lapping and then pacing).

Based upon the above analysis, the sequential tree-based algorithm (*Appendix B*) is formulated. Generally, mapping a single-pattern episode is a special case of mapping a multi-pattern episode. Appendix B1 hence does not distinguish the two cases. To do the classification, the algorithm first checks for direct pattern. If it is indeed a direct pattern, then the episode is concluded as direct. Otherwise, it checks for pacing (line 3), lapping (line 4), and random (lines 5-9) patterns in the current episode and label them accordingly. The concluding pattern is identified based on the number of occurrence or the severity of wandering patterns. The functions isDirect(), checkPacing(), checkLapping() are described in Appendix B2-4.

Algorithms to check for direct, pacing, and lapping

Check for direct and non-direct patterns. A path including two or fewer movements is considered direct. So, Appendix B2 detects a direct episode with more than two movements by checking if there is any repeated location in the episode (line 3) or any shorter or more efficient path that connects the start and end locations (lines 7-10). As mentioned earlier, we look for more efficient path from historical location sequences or episodes so as to rectify the third shortcoming.

Check for pacing patterns. Checking for pacing is done by looking for the repeated pacing sub-pattern (*Appendix B3*), e.g. 'L_1L_2'. The repeated sub-patterns must be continuous. Hence, we have to compare L_i with L_{i+2} and L_{i+1} with L_{i+3} (line 2). If $L_{i+2}L_{i+3}$is the repeated pacing sub-pattern, we sequentially search for all the (continuous) pacing sub-patterns $L_iL_{i+1}$by using the pointer j (lines 3-8). We then label these pacing sub-patterns and update the search pointer i. L_{i+1}can be part of another pacing pattern, hence, we subtract 2 steps from the pointer j.

Check for lapping patterns. By definition, a lapping pattern (e.g., $L_1L_2L_3L_4L_1L_2L_3L_4L_1$ or $L_1L_2L_3L_1L_3L_2L_1$) has its first location (L_1) repeated. To look for a lapping pattern starting from an arbitrary location, we first sequentially search for the repeated locations of the arbitrary location (line 3). Appendix B4 checks for circular paths in the same direction (lines 4-9), e.g. $\underline{L_1L_2L_3L_4L_1L_2L_3L_4L_1}$ and those in opposite directions (lines 17-23), e.g. $\underline{L_1L_2L_3}L_1\underline{L_3L_2L_1}$.We use these example episodes to explain the algorithm.

For lapping in the same direction, once the repeated location (L_1) is found (line 3), the pointers i and j respectively index the first and fifth locations of the location sequence $L_1L_2L_3L_4L_1L_2L_3L_4L_1$. The repeated path (the second sub-sequence $\underline{L_1L_2L_3L_4L_1}$ in the lapping pattern $L_1L_2L_3L_4\underline{L_1L_2L_3L_4L_1}$)is detected by using a step pointer k, which moves through the location sequence to sequentially check for repeated locations (line 6). If there is any location mismatch, it will immediately terminate the while loop (line 7). The value of pointer k therefore indicates the length of the repeated path found. In line 10, we make use of this length to determine that the repeated paths are continuous (to avoid cases such as two repeated paths $\underline{L_1L_2L_3L_4L_1}$ are separated by another sub-pattern of different types, e.g. random). The condition () in line 10 is flexible enough to recognize lapping episodes such as $L_1L_2L_3L_4L_1L_2L_3L_4$ (when the first location is not revisited at last) or $L_1L_2L_3L_4L_1L_5L_1L_2L_3L_4L_1$ (when a single random location L_5 is visited during travel). Line 11 checks if the sub-paths found satisfy the condition of lapping (i.e. they contain at least three distinct locations). The variable tempEnd1 is used to mark the latest location in the episode that has been labeled as lapping. This is to avoid repeated labeling.

In cases of lapping in opposite directions (e.g. $\underline{L_1L_2L_3L_1L_3L_2L_1}$), the length of the location sequence is always an odd number. We use this characteristic to quickly examine if a pattern is a lapping in opposite direction (line 17). We indeed execute the checking for opposite paths (e.g. $\underline{L_1L_2L_3}$ and $\underline{L_3L_2L_1}$) similarly to what we do for detecting lapping in the same directions. However, we search for repeated locations in a reverse direction (line 20). We also use the condition in line 24 to check if the opposite sub-paths are continuous. Finally, we check if the sub-paths found satisfy the conditions for lapping (line 25) and use the variable tempEnd2 to keep track of the latest location that is labeled as lapping.

Table 8. Statistical comparison of the deterministic algorithm and the machine learning approach, using McNemar's test; confidence limit is set at 0.01

Classifier	**McNemar's value**	**p**
NB	26.21	<0.00001
MLP	17.93	0.0002
C4.5	13.07	0.0003
RF	8.64	0.0033
BAG	10.56	0.0012
LB	13.07	0.0003
k-NN	19.05	0.0001
SVM	16.41	0.0005

Table 9. Performance of machine learning approach and deterministic algorithm in classifying direct versus indirect travel; NB= Naïve Bayes.; MLP= Multilayer Perceptron; C4.5= Pruned decision trees; RF= Random Forests; BAG=Bagging; LB=Logiboost ; k-NN=k-Nearest Neighbor; SVM=Support Vector Machine

Classifier	**Sensitivity**	**Specificity**	**Precision**	**Latency**
Machine learning				
NB	84.5	79.6	87.3	0.01
MLP	92.7	93.8	93.2	0.17
C4.5	97.3	96.9	97.3	0.01
RF	98.2	97.8	98.2	0.01
BAG	97.3	97.2	97.3	0.04
LB	97.7	97.5	97.7	0.06
k-NN	93.6	91.7	94.1	0.01
SVM	95.5	95.5	95.5	0.02
Deterministic algorithm	98.6	98.2	98.7	0.0003

The function isLapping checks whether or not the sub-path contains at least three distinct locations by subtracting the length of the location sequence by the number of unique elements or locations in the sequence (*Appendix B5*).

From the analysis of Appendix B2-5, we can see that the complexity of Appendix B1 is Θ (n^2).

Experiment and results

To have a fair comparison with other algorithms, we have applied the deterministic tree-based algorithm on the same test data of the 10-fold cross validation used in the machine learning approach. The sensitivity, specificity, precision, recall, F1-measure, and latency are 98.2%, 98.1%, 98.2%, 98.2%, 98.2%, and 0.0003s respectively. Results show that the deterministic algorithm improves both the classification recall and latency. In particular, the classification recall improves by 5.9% to 98.2% compared to the best result produced by RF. And the latency is reduced remarkably to 0.0003s (100 times better than RF). The deterministic algorithm and the machine learning approach are compared using the McNemar's test[27] to examine if their differences in recall or accuracy are statistically significant (*Table 8*). Differences in classification accuracy of the deterministic algorithm and other eight classifiers are significant ($p<0.01$).

Four misclassified episodes produced by the deterministic algorithm are due to shortcoming 2. The algorithm is not adaptive enough to recognize random patterns such as (CAR2, C5301, AR1, AR2) or (R5304, NS, CAR2, C5309, AR2, C5303). By using location context, it is possible to deduce that the corridors (CAR2, C5303) and the rooms (AR2, C5303) are near to one another. Therefore, such episodes are inefficient travel, not direct travel.

Additionally, we compare the capability of both the machine learning approach and the deterministic tree-based algorithm in the binary classification of direct versus indirect patterns. After all, pacing and lapping patterns are subsets of random patterns with different degrees of randomness or wandering. A simple direct/indirect classification is useful for providing healthcare practitioners a clear and quick assessment of the severity of a subject's wandering behavior.

The binary classification performances are improved for both approaches (*Table 9*). For the machine learning algorithms, the sensitivity values of eight classifiers are improved from 5.6% to 7.8%. Meanwhile, the deterministic algorithm's sensitivity is increased by 0.4% only. Nonetheless, the deterministic algorithm produces the highest sensitivity (98.6%) and specificity (98.2%) with shortest latency. For machine learning approaches, the RF classifier yields the best classification results, which are 98.2% sensitivity, 97.8% specificity, and 0.01s latency. Nevertheless, the difference in performance between the two classifiers is not significant.

Conclusions

In this study, we propose two automated approaches for classifying travel patterns of PWD.

In the machine learning approach, eight different classifiers NB, MLP, LB, C4.5, RF, BAG, k-NN, and SVM are employed. The methodology is evaluated on movement data produced by RFID tags placed on the subjects' body. The subjects in this study are five PWD who have similar age distribution and suffer from wandering. The travel patterns including wandering are manually labelled so as to have a ground truth of classification. The results are expressed in terms of sensitivity, specificity, precision, recall, F1-measure and latency. The sensitivity, specificity, precision, recall, F1-measure, and latency of the RF classification algorithm were 92.3%, 92.3%, 92.2%, 92.3%, 92.2%, 0.03s respectively.

In the deterministic approach, we also introduce a predefined tree-based algorithm that improves the classification accuracy by 5.9% to 98.2% and significantly reduces the classification latency to 0.0003s. The deterministic algorithm offers several advantages: (i) it is not based on thresholds, (ii) it is based on an operationalized (clearly distinguishable and measurable through empirical observations) definition of wandering, (iii) the classification process is formed from a logically deduced state diagram depicting transformations of travel patterns which is agreeable with the empirical results achieved by classical machine learning algorithms, and (iv) the classification latency is very small, which enables the algorithm to be deployed in real-time and mobile applications aiming to detect wandering behavior of PWD as soon as it takes place.

However, our present study has several limitations: (a) it was tested on datasets of five PWD, (b) the movement data are confined by physical rooms and indoor settings. In reality, a subject can wander within a large area such as a function hall or wander outside the care environments, and (c) the algorithm has some shortcomings in the recognition of travel patterns with complicated and varied geometrical forms. Contextual information and geographical information such as the map, layout or design of the monitored area can be incorporated to refine the classification algorithm.

In the future, we have plans to evaluate and validate the proposed algorithms with additional data samples. We will recruit volunteers to act as wanderers so as to collect extra data for our study. Though the data are not of actual PWD, it is probably the second best we can have in the near future. Actual wandering data from PWD is the best and clinically significant data for our study. However, we currently lack the resources and opportunity to collect this source of data. In addition, we would also test the applicability of the proposed deterministic algorithm with finer location data resolution (e.g. 3-D space coordinates).

In order for the proposed logic to work well with finer location data, we suggest to cluster nearby data points (coordinates) together and partition the coordinates location sequences (or trajectories) into sequences of coarse characteristic locations (e.g. areas of 2mx2m). Such tasks can be done by applying the minimum description length principle and density-based spatial clustering of line segments. Then we can apply the proposed logic in the deterministic algorithm to do the classification. We hope to achieve classification results as good as those reported in this pilot study. Additionally, we would like to scale up the deterministic algorithm as the sample size increases by exploring relevant data mining methods geared specifically towards recognizing sequences whose length is not pre-fixed and that have sub-patterns embedded within them such as hidden Markov models or probabilistic suffix tress.

To sum up, wandering is a common behavior in PWD. In clinical and long-term care settings, wandering is traditionally detected by having human observers to observe PWD and document the wandering patterns including pacing, lapping, and random. Such observational method requires labor and time, and lacks objectivity due to the unpredictable nature of the behavior and due to the fact that during short observation period (usually 20 minutes), PWD may appear very well and may not exhibit any wandering behavior. Results from medical studies[4-6] substantiated that increasing amounts of random-pattern wandering (the frequency and overall proportion of time spent) is clearly linked to increased levels of global cognitive impairment. Meanwhile, lapping patterns are less clearly associated with cognitive declination. But, frequent lapping may signify a deteriorating ability to way-finding in PWD[16]. Pacing is not associated with levels of cognitive impairment; however it is more indicative of agitation and anxiety. In addition, there is evidence supporting that early detection of wandering will allow early treatment and potentially slow down the onset of dementia. Kearns et al.[28] found that fractal dimension (a measure of randomness in 14 assisted living facility (ALF) residents' movement paths continuously recorded over 30 days using ultra-wideband sensors) was significantly higher in persons with lower Mini Mental State Examination scores signifying cognitive impairment. In a subsequent study of 28 subjects[29] recorded over one month replicated the findings of Kearns[30] and found that the randomness of the paths taken by elderly ALF residents reliably differentiated PWD from those without the disorder. Most recently Kearns et al.[31] determined that randomness in the paths of 53 ALF residents' walking patterns recorded over a period of one year was strongly related to their imminent fall risk, establishing an important relationship between the random movement patterns in older persons and the risk of falling.

Ultimately, the personalization of PWD's intervention and close observation of the impact of every intervention to the patient's clinical condition is a crucial step for planning a successful treatment strategy. We hope to develop an automated monitoring system that can measure wandering rates, durations and patterns in order to provide early detection of physical and mental changes of PWD. Such an automated system will enable caregivers to arrange better care distributions, treatments and timely interventions for PWD.

Acknowledgement
The authors thank Professor Makimoto Kiyoko, Osaka University, for allowing them to reproduce the figures in Appendix A, originally published in the 11th World Congress on the Internet and Medicine and American Journal of Alzheimer's Disease and Other Dementias.

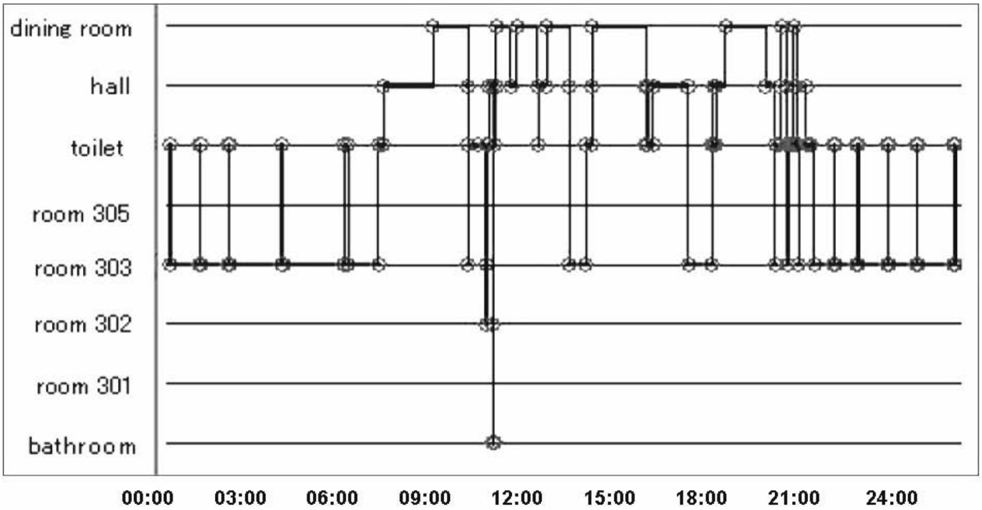
References

1. Martino-Saltzman D, Blasch B, Morris D, McNeal L. Travel behavior of nursing home residents perceived as wanderers and nonwanderers. Gerontologist 1991;31(5):666-672
2. Algase D, Moore H, Vandeweerd C, Gavin-Dreschnack D. Mapping the maze of terms and definitions in dementia-related wandering. Aging & Mental Health 2007;11(6):686-698; doi:10.1080/13607860701366434
3. Algase D, Antonakos C, Beattie E, Beel-Bates C, YaoL. Empirical derivation and validation of a wandering typology. Journal of the American Geriatrics Society 2009;57(11):2037-2045; doi: 10.1111/j.1532-5415.2009.02491.x
4. Algase D, Beel-Bates C, Beattie E. Wandering in long-term care. Annals of Long-Term Care, Clinical Care and Aging 2003;11(1):33-39
5. Lai C, Arthur D. Wandering behavior in people with dementia. Journal of Advanced Nursing 2003;44(2):173-182; doi:10.1046/j.1365-2648.2003.02781.x
6. Rolland Y, Payoux P, Lauwers-Cances V, Voisin T, Esquerré JP, Vellas B. A SPECT study of wandering behavior in Alzheimer's disease. International Journal of Geriatric Psychiatry 2005;20(9):816-820; doi:10.1002/gps.1362
7. Peatfield J, Futrell M, Cox C. Wandering: an integrative review. Journal of Gerontological Nursing 2002;28(4):44-50
8. Wang PN, Yang CL, Lin KN, Chen WT, Chwang LC, Liu HC. Weight loss, nutritional status and physical activity in patients with Alzheimer's disease. Journal of Neurology 2004;251(3):314-320
9. Lee H, Algase D, McConnell E. Relationship between observable emotional expression and wandering behaviour of people with dementia. International Journal of Geriatric Psychiatry 2014;29(1):85-92; doi:10.1002/gps.3977
10. Algase D, Antonakos C, Beattie E, Hong G, Beel-Bates C. Are wandering and physically nonaggressive agitation equivalent? American Journal of Geriatric Psychiatry 2008;16(4):293-299; doi:10.1097/JGP.0b013e3181629943
11. Wimo A, Prince M. World Alzheimer Report 2010: The global economic impact of dementia. Alzheimer's Disease International, 2010
12. Profiling the dementia family carer in Singapore; www.alz.org.sg/sites/alz.org.sg/files/Profiling%20the%20Dementia%20Family%20Carer%20in%20Singapore%20-%20Executive%20Summary.pdf; retrieved October 20, 2013
13. Angiullo L. Wandering behavior in people with dementia. PhD thesis, University of Massachusetts, 1997
14. Kearns W, Rosenberg D, West L, Applegarth S. Attitudes and expectations of technologies to manage wandering behaviour in persons with dementia. Gerontechnology 2007;6(2):89-101; doi:10.4017/gt.2007.06.02.004.00
15. Kearns W, Fozard J. Ultra-wideband radio assessment of locomotion in elderly. Gerontechnology 2008;7(2):138; doi:10.4017/gt.2008.07.02.075.00
16. Makimoto K, Yamakawa M, Ashida N, Kang Y, Shin K. Japan-Korea joint project on monitoring people with dementia. 11th World Congress on the Internet and Medicine; 2006
17. Greiner C, Makimoto K, Suzuki M, Yamakawa M,

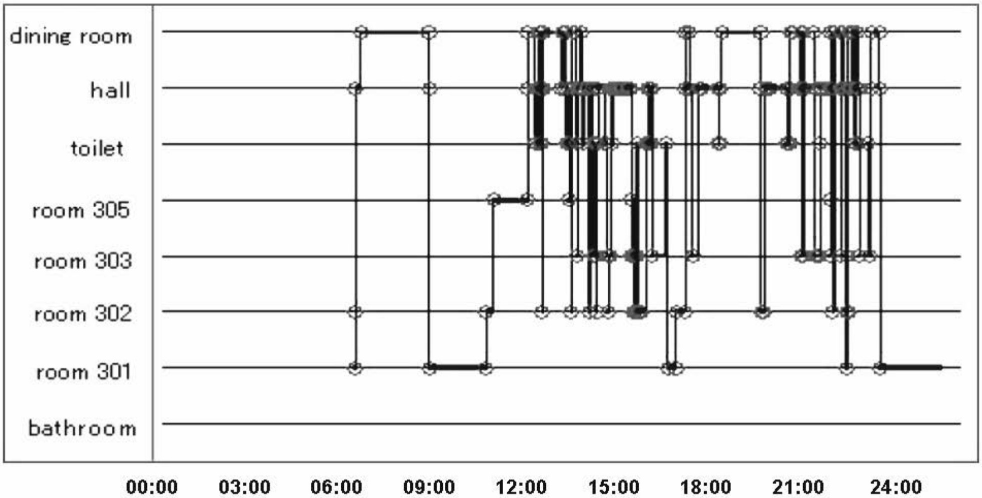
Ashida N. Feasibility study of the integrated circuit tag monitoring system for dementia residents in Japan. American Journal of Alzheimer's Disease and Other Dementias 2007;22(2):129-136; doi:10.1177/1533317507299414.
18. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I. The WEKA Data Mining Software: An Update. SIGKDD Explorations 2009;11:1
19. Rish I. An empirical study of the naïve Bayes classifier. IJCAI-01 Workshop on Empirical Methods in AI; 2001
20. Mitchell T. Machine Learning. New York: McGraw Hill; 1997
21. Quinlan J. Induction of decision trees. Machine learning 1986;1(1):81-106
22. Darnall ND, Donovan CK, Aktar S, Tseng HY, Barthelmess P, Cohen PR, Lin DC. Application of machine learning and numerical analysis to classify tremor in patients affected with essential tremor or Parkinson's disease. Gerontechnology 2012;10(4):208-219; doi:10.4017/gt.2012.10.4.002.00.
23. Breiman L. Random forests. Machine Learning 2001;45(1):5–32
24. Friedman J, Hastie T, Tibshirani R. Additive Logistic Regression: a Statistical View of Boosting. Stanford; Stanford University; 1998
25. Breiman L. Bagging predictors. Machine Learning 1996;24(2):123-140
26. Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers. Proceedings of the 5th annual workshop on computational learning theory. Pittsburgh: ACM; 1992; pp 144-152
27. Fleiss J. Statistical Methods for Rates and Proportions, 2nd edition. New York: Wiley; 1981
28. Kearns W, Nams V, Fozard J. Tortuosity in movement paths is related to cognitive impairment. Wireless fractal estimation in assisted living facility residents. Methods of Information in Medicine 2010;49(6):592-598; doi:10.3414/ME09-01-0079
29. Kearns W, Fozard J, Nams V, Craighead J. Wireless telesurveillance system for detecting dementia. Gerontechnology 2011;10(2):90-102; doi:10.4017/gt.2011.10.2.004.00
30. Kearns W, Fozard JL, Becker M, Jasiewicz JM, Craighead JD, Holtsclaw L, Dion C. Path tortuosity in everyday movements of elderly persons increases fall prediction beyond knowledge of fall history, medication use, and standardized gait and balance assessments. Journal of the American Medical Directors Association 2012;13(7):665.e7-665.e13; doi:10.1016/j.jamda.2012.06.010.
31. Ashida N, Makimoto K, Nozaki C, Nakagawa M, Suzuki M. A monitoring of wandering behavioural pattern of senile dementia with IC tag. 3rd APT Telemedicine, 2005, Kuala Lumpur, Malaysia, pp 80-84

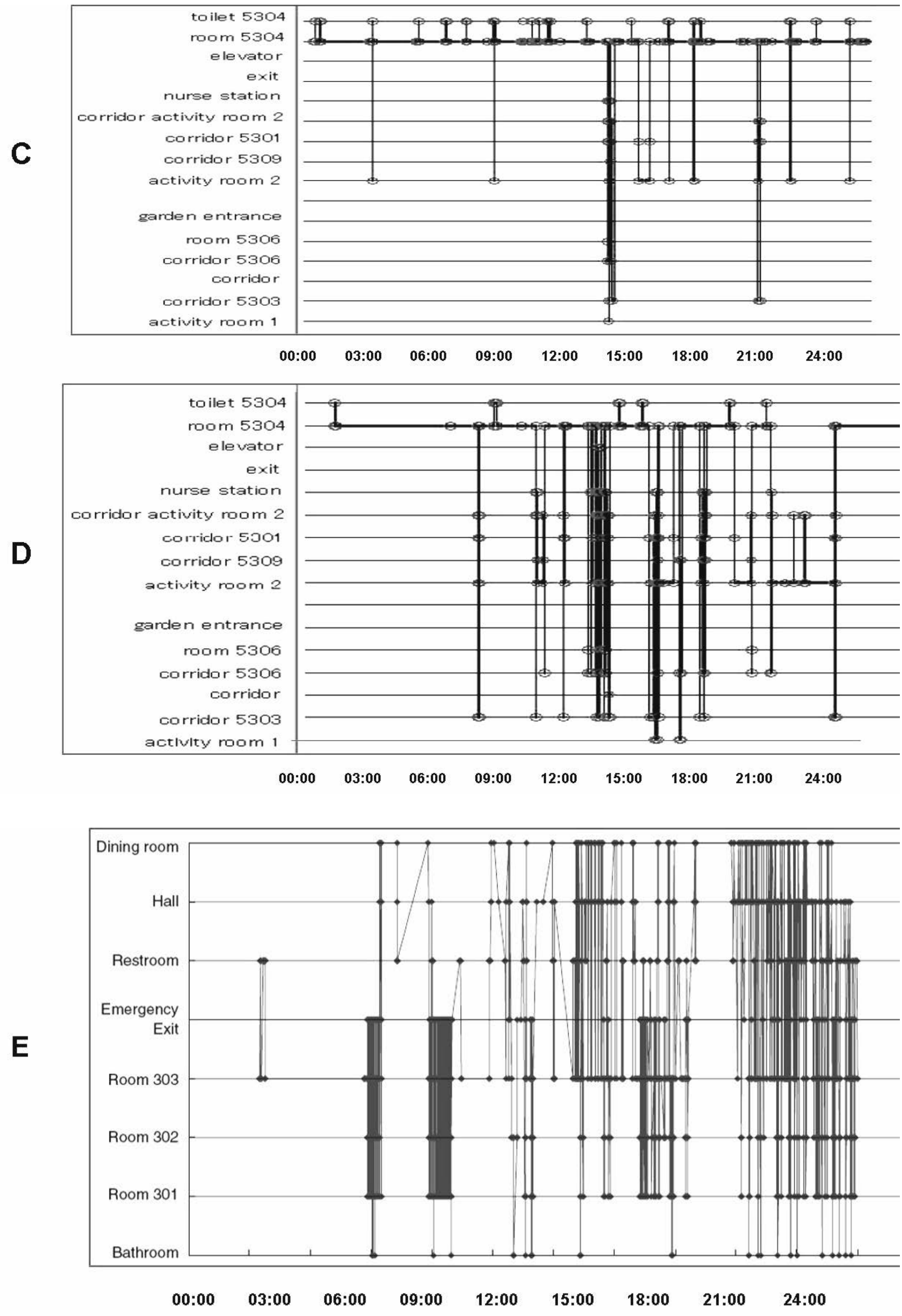
Appendix A.
Movement records of subjects A to E[17]; location and time data recorded by the monitoring system[16] are presented in the vertical and horizontal axis respectively

A



B




C
toilet 5304
room 5304
elevator
exit
nurse station
corridor activity room 2
corridor 5301
corridor 5309
activity room 2
garden entrance
room 5306
corridor 5306
corridor
corridor 5303
activity room 1
00:00 03:00 06:00 09:00 12:00 15:00 18:00 21:00 24:00
D
toilet 5304
room 5304
elevator
exit
nurse station
corridor activity room 2
corridor 5301
corridor 5309
activity room 2
garden entrance
room 5306
corridor 5306
corridor
corridor 5303
activity room 1
00:00 03:00 06:00 09:00 12:00 15:00 18:00 21:00 24:00
E
Dining room
Hall
Restroom
Emergency Exit
Room 303
Room 302
Room 301
Bathroom
00:00 03:00 06:00 09:00 12:00 15:00 18:00 21:00 24:00

APPENDIX B.

Deterministic tree-based algorithms for classifying travel patterns

B1. Classification of Wandering Patterns

Inputs:
- $L_1, L_2, \dots, L_n (n \in \mathrm{N})$: sequence of previously visited locations;

Output:
- pattern type ("direct", "random", "lapping", or "pacing") of ambulation $L_1, L_2, \dots, L_n$

```
if (isDirect($L_1, L_2, \dots, L_n$)) label "direct" for the episode containing $L_1, L_2, \dots, L_n$;
else
    checkPacing($L_1, L_2, \dots, L_n$) and label "pacing" for corresponding sub-patterns;
    checkLapping($L_1, L_2, \dots, L_n$) and label "lapping" for corresponding sub-patterns;
    for each remaining unlabeled sub-sequence $S_{ij}(L_i, \dots, L_j)(1 \le i \le j \le n)$
        if (isDirect($L_i, \dots, L_j$)) label "direct" for $S_{ij}$;
        else label "random" for $S_{ij}$;
        endif
    endfor
    Ra, La, Pa = the number of sub-patterns labeled as "random", "lapping", and "pacing" respectively;
    $max = Max(Ra, La, Pa)$;
    if ($max == Ra$) label "random" for the episode containing $L_1, L_2, \dots, L_n$;
    else if ($max == La$) label "lapping" for the episode containing $L_1, L_2, \dots, L_n$;
    else label "pacing" for the episode containing $L_1, L_2, \dots, L_n$;
    endif
endif
```

B2. isDirect($L_1, L_2, \dots, L_n$) : check if a travel pattern is of "direct" type

Inputs:
- $L_1, L_2, \dots, L_n (n \in \mathrm{N}, \mathrm{n} > 3)$: sequence of previously visited locations;

Output:
-**true** or **false**: whether the pattern is "direct"

```
for $i = 1{:}n$-1
    for $j = i + 1{:}n$
        if ($L_i == L_j$ )returnfalse;
        endif
    endfor
endfor
search for paths containing $L_1$ and $L_n$ from previous location sequences
    if there is any more efficient path connecting $L_1$ and $L_n$ than the current path ($L_1, L_2, \dots, L_n$)
        return false;
    endif
return true;
```

B3. checkPacing($L_1, L_2, \dots, L_n$) : search and label "pacing" if exist.

Inputs:
- $L_1, L_2, \dots, L_n (n \in \mathrm{N})$: sequence of previously visited locations;

Output:
- label "pacing" for "pacing" patterns

```
for $i = 1{:}n - 3$
    if($L_i == L_{i+2} \&\& L_{i+1} == L_{i+3}$)
        $j = i + 4$;
        while($j \le n - 1$)
            if($L_i == L_j \&\& L_{i+1} == L_{j+1}$)$j$ += 2;
            elsebreak;
            endif
        endwhile
        label "pacing" for $L_i \dots L_{j-1}$;
        i = j – 2;
    endif
endfor
```

B4. checkLapping($L_1, L_2, \ldots, L_n$): search and label "lapping" if exist.

Inputs:
- $L_1, L_2, \ldots, L_n (n \in \mathrm{N})$: sequence of previously visited locations;
- $stack$: an empty stack data structure;
- $tempEnd1 = -1$;
- $tempEnd2 = -1$;

Output:
- label "lapping" for "lapping" patterns

```
for i = 1:n-1
    for j = i + 1:n
        if (L_j == L_i)
            k = 0;
            while(i + k ≤ j && j + k ≤ n)
                if(L_{i+k} == L_{j+k}) k += 1;
                else break;
                endif
            endwhile
            if (i + k == j || i + k == j − 1)
                if (tempEnd1 < j + k && isLapping(L_i ... L_{j−1}))
                    label "lapping" for L_i ... L_{j+k−1};
                    tempEnd1 = j + k;
                    break;
                endif
            endif
            if((j − i)%2 == 0)
                k = i + 1;
                while(k < (i + j)/2 && k ≤ n)
                    if(L_k == L_{j−k+i}) k += 1;
                    else break;
                    endif
                endwhile
                if (k == (i + j)/2)
                    if(tempEnd2 < j && isLapping(L_i ... L_{j−1}))
                        label "lapping" for L_i ... L_{j+k−1};
                        tempEnd2 = j;
                        break;
                    endif
                endif
            endif
        endif
    endfor
endfor
```

B5. isLapping($L_i \ldots L_{j-1}$): check if sub-sequence $L_i \ldots L_{j-1}$ contains at least 3 distinct locations

Inputs:
- $L_i, \ldots, L_{j-1} (n \in \mathrm{N})$: sub-sequence to check;

Output:
- **true** or **false**: whether it contains at least 3 distinct locations

```
if (j − i − uniqueElementsof(L_i, L_{i+1}, ... L_{j−1}) < 3) return false;
return true;
```