Sensors and Materials, Vol. 34, No. 8 (2022) 3029–3045
MYU Tokyo

S & M 3021

Indoor Localization under Sparse Bluetooth Low Energy Scanner Deployment for Medical and Nursing Care of Elderly People

Takuya Yoshihiro* and Takeshi Goya

Faculty of Systems Engineering, Wakayama University, 930 Sakaedani, Wakayama 640-8510, Japan

(Received March 31, 2022; accepted May 31, 2022)

Keywords: indoor localization, BLE beacon, elderly care, map matching, travel route estimation

Disappearances and falls are problems in the medical and nursing care of elderly people. There is also a shortage of nurses and caregivers, which increases the workload per person. A system that automatically monitors the movements of elderly people and alerts nurses and caregivers of emergencies would make it possible to solve problems safely and efficiently even with a small number of staff. In this study, we propose a system in which a small Bluetooth Low Energy (BLE) tag is continuously worn by an elderly person and BLE scanners are installed at various locations in medical and nursing care facilities to estimate their routes in real time with sufficient accuracy for practical use. To achieve this, several constraints must be overcome. First, the BLE beacon tags must be small enough to be worn. Second, since a tag must have relatively long battery life, we cannot avoid long-cycle and low-power beacon transmissions. Third, the locations where BLE scanners can be placed in a building are limited, resulting in their sparse deployment. Under these assumptions, we propose graph-based indoor localization for sparse scanner deployment (GILS) as a method that estimates indoor locations and movement paths of elderly people by incorporating the map matching technique in real time. By introducing heuristics to correct locations in real-time location tracking, we achieve a practical level of accuracy in location and movement path estimation even with sparse beacon transmission and scanner deployment.

1. Introduction

In recent years, the population of many countries has been aging and the proportion of elderly people has been increasing, especially in Japan.[1,2] As a result, hospitals and elderly-care facilities are seeing an increase in the number of inpatients and residents, leading to a shortage of nursing and care workers. Medical and nursing care facilities are also experiencing problems such as elderly people wandering outdoors and going missing, or becoming ill in the middle of the night in restrooms, hallways, and other facilities and being unable to move because they cannot call for help. If these problems are not detected early, they can be life threatening. The

*Corresponding author: e-mail: tac@wakayama-u.ac.jp
https://doi.org/10.18494/SAM3909

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

problem of going missing outdoors can be prevented if staff know when an elderly person tries to leave the house and take immediate action, and an elderly person who is unable to move or call for help can be detected early if their location is tracked.

One possible solution to these problems is to introduce a system that tracks the location of elderly people and detects their problems in real time. The system notifies mobile devices carried by a nurse or caregiver when an elderly person tries to go out of the house or is immobilized at a location for a long time. The location of the elderly person is then displayed on the mobile devices.

Although many studies on location estimation exist,[(3)] there are several problems with its use in medical or elderly-care scenarios. First, if elderly people need to wear a device (or a tag) continuously, it must be sufficiently small for them to forget they are wearing it, which limits such a device to a Bluetooth Low Energy (BLE) beacon at the current level of technology. Since it is costly to replace the battery frequently, the transmission power of the small beacon tag must be small and the transmission cycle must be long. In this case, location estimation is based on a small number of weak beacons, and the accuracy of location estimation is significantly reduced. On the other hand, the BLE scanner that receives the radio waves emitted by the small tags is also limited in where it can be installed within a building. The cost of scanners is also an issue, and their placement must be somewhat sparse. Consequently, we need a method to maintain the necessary position estimation accuracy even if BLE beacons are received by a small number of scanners.

To address this issue, we propose graph-based indoor localization for sparse scanner deployment (GILS) as a method to estimate the locations of elderly people with sufficient accuracy for practical use, even when the scanner placement is sparse and the beacon time interval is long. First, in this study, we use Raspberry Pi for a BLE scanner and equip it with four additional BLE interfaces to obtain five received signal strength identifier (RSSI) values from a single beacon. It has been observed that RSSI values obtained from multiple BLE adapters vary even for the same beacon, contributing to improved location estimation accuracy. To prevent position estimation accuracy from degrading even when the number of scanners capable of receiving each beacon is small, a heuristic based on a map is applied to adjust the estimated location in real time using the time-series consistency of the estimated location. Specifically, we define a graph that defines the possible locations on a map, estimate the location using RSSI values, and adjust the location on the basis of time-series consistency to estimate the route of elderly people in real time. Through evaluation, we show that the proposed method GILS achieves practical performance.

The structure of this paper is as follows. Section 2 describes related work on estimating locations under sparse deployment of BLE beacons. Section 3 describes the framework of the proposed system. Section 4 describes the proposed location estimation method of GILS. Section 5 evaluates the accuracy of the proposed method, and finally, Section 6 summarizes the contributions of this study.

2. Related Work

There have been many studies on indoor location estimation, and methods based on several strategies such as channel state information, fingerprinting, angle of arrival, and time of arrival have been proposed. [3] Among them, the most basic method is based on the RSSI. For this strategy, various radio sources can be used, including Wi-Fi, Bluetooth, and cellular phones. However, Wi-Fi and Bluetooth are most practical because these standards are popular in indoor use.

For indoor localization in medical and elderly-care applications, Wi-Fi and Bluetooth are suitable from the viewpoint of cost. Since elderly people must wear tags continuously, Bluetooth is considered the most promising method in terms of low power consumption. Localization using Bluetooth is a well-investigated area of study, which includes methods using RSSI-based distance estimation[4] and deep learning.[5] However, these methods assume that scanners are densely placed in buildings, which does not apply to general medical and elderly-care settings because the locations where BLE scanners can be placed are severely restricted. Furthermore, we also have a severe limitation on the time interval and transmission power of BLE beacons because of the small battery capacity of small tags worn continuously by elderly people, which is also not acceptable in these methods. Conversely, there is a system design where beacon tags that can operate for a long period of time with a small battery capacity are densely placed in buildings, and elderly people carry BLE scanners with them. This setting may be effective for estimating their location with high accuracy, and such location estimation methods using fingerprinting[6] and dead reckoning[7] have been proposed. However, it is difficult for elderly people to carry large electronic devices such as smartphones with them continuously, and even relatively small beacon tags are often left behind in their rooms.

On the other hand, to complement the low accuracy of location estimation by GPS, the map matching technique, which uses a map to restrict the possible locations, has been proposed.[8,9] Also, maps have been combined with gyro sensors without using other localization methods such as GPS.[10] Yamamoto *et al.* applied the global matching method to localization with BLE beacons,[11] and global matching has been practically applied to cow tracking.[12] Yamamoto *et al.*[11] represented a map by a graph, where vertices represented movable positions and edges represented possible transitions. They computed the time series of locations (i.e., vertices) with the highest likelihood given the time series of RSSI values. Although this method considers the case where BLE scanners are sparsely located, it computes the most likely routes after the whole data set is obtained; thus, it is not possible to compute the location in real time. Since we need to detect problems in real time to generate alerts and inform nurses, their method cannot be applied. In addition, it does not consider the case of a time interval in which no scanner receives beacons, meaning that we must consider the sparser deployment of scanners; we consider the case where a beacon is often received by less than two scanners. In such cases, we sometimes must estimate locations based on proximity,[13] and in the worst case, there is no scanner that can receive beacons. In this paper, we propose a BLE-based real-time localization method for such sparse deployment of BLE scanners.

3. Framework of Proposed System

3.1 Requirements of medical and elderly-care facilities

In hospitals and nursing homes, elderly people often stay for long periods of time, and automatic detection of their problems is required to improve the efficiency of their care. In hospitals, inpatients may leave the hospital without permission and, especially in cases of dementia, there is a risk that they may go missing. Although there are commercially available devices (e.g., anti-theft gates) that notify nurses when a beacon tag passes a certain location, they lack functionality because elderly people sometimes leave their beacon tags in their rooms. In many nursing homes in Japan, doors to the outside are locked and elderly people cannot leave without permission. However, there is a risk of falling when getting up from bed, getting into a wheelchair, or moving around. This means that care workers need to know when elderly people make certain movements. These requirements can be satisfied if the indoor location of an elderly person can be always monitored and the nurse or caregiver can be notified when necessary.

To keep track of the location of elderly people indoors, we first need them to carry a beacon tag continuously. Since the situation depends on the hospital or care facility, this should be done in a way suitable for each facility. If an elderly person always wears a name tag, it is easy to attach a beacon to it. However, in many cases, name tags are not worn. In some hospitals, people are identified by wearing a tag with a bar code printed on it around the wrist. In this case, a beacon tag can be attached to this tag, but it must be small. If neither a name tag nor a wrist tag is available, a small beacon tag should be attached to a shoe or another object.

On the other hand, beacon tag batteries are a major problem in actual operation. Frequent battery replacements are not acceptable for business operations, so it is preferable to have batteries that can operate for as long as possible without replacement. However, a small size generally means a small battery capacity, and the only way to balance size and battery capacity is to lengthen the time interval of the beacon transmission. In this study, Mamorio[14] was selected as the beacon tag to balance these factors. This small beacon tag has dimensions of 19 × 35 × 3.4 mm and a battery life of approximately one year, and it transmits beacons with 75 dB power at intervals of about 3 s.

BLE beacon scanners that receive beacons must be installed in the building. In general, however, scanners can only be installed where there is a power source. In addition, from the standpoint of financial cost, no more than the necessary number of scanners should be installed. Therefore, we assume a sparse scanner arrangement where it is difficult to always capture a beacon with three or more scanners, and we place additional scanners in areas where accurate position estimation is particularly necessary. In other words, in the medical care field, it is necessary to achieve sufficient position estimation accuracy to determine the path of movement under the assumption of long beacon intervals and sparse scanner placement.

We also have requirements on time delay and accuracy in the medical and elderly care field. Although we wrote that real-time localization is required, a certain delay of a few minutes is accepted in this field. In most cases, trouble with elderly people is solved if someone can arrive within a few minutes. On the other hand, the requirement for localization accuracy depends on the case. However, since we define the possible locations of elderly people as the route graph associated with the indoor map, the localization results will be applicable in practice if the route graph is appropriately designed according to the required granularity. Appropriate placement of BLE scanners with respect to the route graph will provide an accurate path estimation and low localization error. To achieve this is the objective of this paper to satisfy the practical requirement.

3.2 Framework of the proposed system

An overall view of the proposed system to be used in hospitals and elderly-care facilities is shown in Fig. 1. The system consists of four elements: a BLE beacon tag, BLE scanners, a server, and mobile devices. A BLE scanner is typically implemented on a small device such as Raspberry Pi and is often installed near a power outlet. BLE scanners should be placed in a manner such that the movement history is well obtained, i.e., the system can estimate the position with the required accuracy. When a scanner receives a beacon, it forwards the record to a server. The server can be located either at the facility in which scanners are placed or outside the facility. The beacon reception history is stored in a database in the server. Mobile terminals such as smartphones or tablets are provided for use by nurses and caregivers and are placed at the room where they usually work. Alternatively, each person may carry a mobile phone to receive alerts from the system. If the server detects any problems, it sends an alert to the tablet or smartphone via push communication to inform the nursing staff or caregiver. If necessary, the server notifies the device with detailed descriptions such as the location with a map and the recent moving history. The person who receives the notification can respond promptly to the problem by taking action according to its contents.

3.3 Equipping multiple BLE adapters on a scanner

As mentioned earlier, to meet the requirements of small size and long life, the beacon tags used in the field require a long beacon transmission cycle and low transmission power, which reduces the accuracy of position estimation. To address this problem, we connect multiple BLE adapters to the scanner. In general, Raspberry Pi is often used as a BLE scanner, which has four USB ports. By connecting external BLE adapters to these ports, five BLE adapters including the built-in BLE adapter can be installed. Figure 2 shows a photograph of the prototype system. The

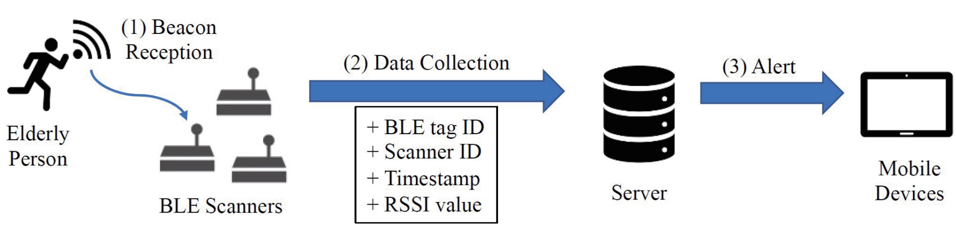

Fig. 1. (Color online) Overview of the proposed system.

Fig. 2. (Color online) Scanner equipped with four BLE adapters.

Raspberry Pi in the foreground is the scanner with four BLE adapters installed. Four is the maximum number due to the limitation of the number of USB ports, although a larger number of adapters will improve the localization accuracy.

4. Proposed Method GILS for Sparse Scanner Deployment

4.1 Overview

We propose a method for estimating indoor locations and moving paths with practical accuracy for a sparse scanner arrangement and long beacon transmission cycle. By connecting multiple BLE adapters to a beacon scanner, multiple RSSI values are acquired and averaged for a single beacon, thereby reducing the error in the distance estimated from the RSSI values. This reduces the error caused by the sparse scanner placement. The proposed method restricts the movement paths on a graph defined on an indoor map, i.e., we assume that people move only on the edge of the given route graph. The distance from the scanner is estimated from the RSSI values of the received beacons to limit the possible locations on the moving path. We first compute the estimated distance from each scanner from a set of RSSI values measured by the scanner, and determine several possible locations on the route graph. After that, we check the consistency of the estimated locations in a time series to identify the route of the tag's movement. Although accurate location estimation is generally not possible unless three or more scanners receive beacons, we achieve practical accuracy in location estimation by limiting the possible locations using the route graph and by checking the consistency of the estimated locations in the time series.

Specifically, in GILS, we first set time windows at regular intervals and estimate the positions on the route graph for each time window using the RSSI values of beacons observed by scanners. If the number of scanners that have observed beacons is not sufficient to determine a unique estimated location, multiple estimated locations are left as a set of estimated locations. Next, the estimated location sets in the time series are checked, and inconsistent estimated locations are deleted by moving backward in time. If a beacon is not observed in a time window, the position in the time window is added using the locations estimated before and after that time window. By repeating the above steps, a unique estimated movement path is finally identified.

4.2 Data set

The input of GILS is the route graph set on the indoor map and a time series of beacon reception records obtained by scanners. The route graph $G = (V, E)$ is geographically matched to the indoor map, which represents locations that the tags can move to. Namely, each vertex $v \in V$ has a coordinate on the map, and we assume that tags can move on edges $e \in E$.

The beacon reception records of a scanner are a set of tuples generated each time a beacon is received by a BLE adapter attached to the scanner. Each tuple includes the reception time, scanner ID, beacon tag ID, BLE adapter ID, and RSSI value. In this study, each scanner is equipped with five BLE adapters, so for each beacon sent, each scanner receiving it generates up to five tuples. A time window of a few seconds is set as the unit for estimating the position. We let the period at which the time window starts be W_s (s) and the length of the time window be W_w (s). That is, the start time of time window $t = 1, 2, \ldots$ is tW_s and the end time is $tW_s + W_w$.

For each time window, we pre-process the data. We collect a set of tuples where a BLE adapter of scanner s receives a beacon of beacon tag b in time window t. If the number of tuples is less than or equal to three, we delete them. Otherwise, we take the average of the RSSI values, which is added to the input data set. As a result, data set D_t for time window t contains at most one average RSSI value for each pair of scanner s and beacon tag b.

4.3 Algorithm to estimate locations

Our proposed algorithm (Algorithm 1) focuses on a single target beacon tag to predict locations. This algorithm is executed each time the data set of time window t, denoted by D_t, is obtained. In line 1, we retrieve D_t from D, which is the family of data sets of all time windows in the past. In line 2, we execute step (1) to obtain the set of estimated locations L_t of time window t. In line 3, we insert L_t into L, which is the family of location sets of all time windows in the past. Finally, we execute step (2) to check the consistency of the estimated locations and estimate the moving path of the target tag.

4.3.1 Procedure of step (1)

Function 1 describes step (1) of the proposed method. Line 2 substitutes the number of scanners that have received beacons with the target tag in time window t according to data set D_t. The procedure branches according to the value of x. In the case of $x \geq 3$ (line 3), we first calculate the optimal coordinate l' with the minimum square error according to the RSSI values of the multiple scanners included in D_t (line 4). Specifically, for each RSSI value of the scanners, we estimate the distance between the scanner and target tag using Friis's equation,[(15)] and compute the optimal coordinate that minimizes the error in the distance. Since this coordinate may not be on an edge of graph G, we compute the nearest coordinate l on G in line 5 and substitute it as the set of estimated locations L_t in line 6.

In the case of $x = 2$ (line 8), there are two scanners that have received beacons from the target tag. Thus, in line 9, we first compute the two intersections of the two circles with diameters

Algorithm 1
Indoor Localization Algorithm

```
Dt ← D.retrieve(t)              ▷ retrieve tuples for time window t
Lt ← computeLocationSet(Dt)     ▷ obtain location set for t
L ← L ∪ Lt                      ▷ insert into location set family L
checkLocations(L, t)
```

Function 1
Step (1): Estimating Locations for Time Window

```
1  function computeLocationSet(Dt)
2      x ← Dt.numOf Scanners()
3      if x ≥ 3 then
4          l′ ← Dt.calcMinSqrLoc()
5          l ← G.findLocOnGraph(l′)
6          Lt ←{l}
7      end if
8      if x = 2 then
9          L′t ← Dt.findCrossPoints()
10         if L′t = Ø then
11             l′t ← Dt.calcMinSqrLoc()
12             L′t ← {l′t}
13         end if
14         Lt ← G.findLocOnGraph(L′t)
15     end if
16     if x = 1 then
17         d ←  Dt.estimateDistance()
18         Lt ← G.findClosestLocOnGraph(d)
19     end if
20     if x = 0 then
21         Lt ← Ø
22     end if
23 end function
```

corresponding to the RSSI values of the two scanners as a set of estimated locations L'_t. If L'_t is empty (line 10), i.e., the two circles do not intersect, we compute the coordinate with the minimum square error that is on the line connecting the two coordinates of the scanners in line 11 and assign it as a set of estimated locations L'_t (not on graph G). In line 14, we compute the nearest locations on G for each element of L'_t to obtain a set of estimated locations L_t on G.

In the case of $x = 1$ (line 16), we first calculate the estimated distance d from the RSSI value (in line 17) and find the coordinates on G that are the closest to distance d (in line 18). Note that if multiple edges on G intersect the circle, all of them are estimated locations L_t.

Finally, if $x = 0$ (line 20), we have no information to estimate the location for time window t. Thus, L_t should be an empty set (line 21).

4.3.2 Procedure of step (2)

Function 2 describes step (2) of the proposed method. This procedure checks the family of estimated location sets backward in time, removing inconsistent locations or filling in missing

Function 2
Step 2: Identifying Routes

```
1  function checkLocations(L, t)
2  f_trackback ← 1
3      while f_trackback = 1 do
4          f_trackback ← 0
5          if |L_t| = 0 then break end if            ▷ Skip processing t if no location
6          /* if L_{t−1} is empty, complement the past location */
7          if |L_t| > 0 and |L_{t − 1}| = 0 then
8              k ← t − 1
9              while L_k = 0 do k ← k − 1 end while
10             (l_t, l_k) ← G.findNearestPairOnGraph(L_t, L_k)
11             if dist(l_t, l_k) ≤ δ × (t − k) then
12                 L.complementLocation(k, t)
13             end if
14             L_k ← {l_k}
15             t ← k
16:            f_trackback ← 1
17         end if
18         /* if l_t is unreachable from none of L_{t − 1}, remove it */
19         foreach l_t ∈ L_t do
20             if L_{t − 1}.noLocWithinDistance(l_t, δ) then
21                 L_t.remove(l_t)
22             end if
23         end foreach
24         /* if l_{t − 1} is unreachable from none of L_t, remove it */
25         foreach l_{t − 1} ∈ L_{t − 1} do
26             if L_t.noLocWithinDistance(l_{t − 1}, δ) then
27                 L_{t − 1}.remove(l_{t − 1})
28:                f_trackback ← 1
29             end if
30         end foreach
31         /* if |L_{t − 1}|(> 2) will not decrease, choose one. */
32         if |L_t| = 1 and L_{t − 1} ≥ 1 then
33             (l_t, l_{t − 1}) ← G.findNearestPairOnGraph(L_t, L_{t − 1})
34             L_{t − 1} ← {l_{t − 1}}
35             f_trackback ← 1
36         end if
37         t ← −1
38     end while
39 end function
```

locations to uniquely identify the travel route of the tag. Line 2 initializes the flag *fbacktrack*, implying the need to check the estimated locations further back in the past. In line 3, if the flag is set, time window t is processed, and the flag is reset in line 4. In line 5, if L_t is empty, it cannot be checked and the while loop is broken.

In lines 6–17, we complement the past locations if $|L_t| > 0$ and $|L_t - {}_1| = 0$. Lines 7–9 find time index k where $|L_t| > 0$ and $|L_k| > 0$ but $L_k, L_{k+1}, \ldots, L_{t-1}$ are all empty. Namely, we complement the locations of $L_k, L_{k+1}, \ldots, L_{t-1}$. Then, we find the pair of locations (l_k, l_t) where $l_k \in L_k$, $l_t \in L_t$, and the distance between l_k and l_t is the smallest in line 10. We assume that a tag can move a distance between two adjacent time windows. Thus, if the distance between l_k and l_t is less than $\times (t - k)$ (line 11), we complement $l_{k+1}, \ldots, l_{t-1}$ as if the tag has moved from l_k to l_t along the shortest path on G at a constant speed (line 12). In this case, we determine the location for time window k uniquely as l_k (line 14). After substituting k for t, since L_k is modified, we should check further back in the past and set the flag $f_{trackback}$ (line 16). Otherwise, i.e., if the distance between l_k and l_t is larger than $\delta \times (t{-}k)$ in line 11 and we cannot complement locations, then we carry out the action as in lines 14–16 because we do not subsequently check time window k.

In lines 18–30, we check the distance between the estimated locations of adjacent time windows t and $t-1$. First, we check locations in L_t in lines 19 – 23. If location $l_t \in L_t$ is further than from all locations in L_{t-1}, we delete l_t from L_t since l_t is not reachable. Next, we check locations in L_{t-1} similarly in lines 25 – 30. If a location $l_{t-1} \in L_{t-1}$ is further than from all locations in L_t, we delete l_{t-1} from L_{t-1} since l_{t-1} is not reachable. Once L_{t-1} is modified, we set flag $f_{trackback}$ in line 28 to check the effect on the past.

In lines 31–36, we reduce the number of estimated locations to one when necessary. If $|L_t| = 1$, there is no possibility in the future to remove elements of L_{t-1} upon acquiring more information. Thus, in this case, we choose the most likely location in L_{t-1} and remove the others. Specifically, we retrieve the nearest pair (l_t, l_{t-1}) in line 33, and remove locations of L_{t-1} other than l_{t-1} in line 34. Since L_{t-1} is modified, we set flag $f_{trackback}$ in line 35. Finally, we decrease t in line 37 and execute the while loop again if $f_{trackback}$ is set.

5. Evaluation

5.1 Methods

The proposed system was installed in a nursing home in Wakayama City, Japan. We moved around the building with a BLE tag for 10 min to obtain a data set. After applying preprocessing as mentioned in Sect. 4.2, we evaluated the performance of GILS. In our evaluation, Mamorio Fuda shown in Fig. 3(a) was used as BLE tag, which is a tiny sticker-type BLE beacon of size 24 × 36.2 × 3.4 mm^3 and weight 3.4 g. [Fig. 3(b) is a same-size black tag put on a wristband, on which a barcode is usually printed.] For a beacon transmission cycle of 3 s and transmission power of 75 dB, the battery life is approximately 1 year. We implemented BLE scanners on Raspberry Pi 3 Model B+ with four additional Elecom LBT-UAN05C1 BLE adaptors, as shown in Fig. 2. Figure 4 shows the indoor map of the nursing home with the route graph and the locations of scanners. Note that the upper part of the map around Nodes A, C, D, E, F, and I is a partition-free hall where elderly people can eat, watch TV, talk, read books, and so forth. The size of the floor is about 80×50 m. We set 19 scanners in the floor, with scanners 1 and 2 in the nurse room, scanners 3, 8, and 9 in the corridor, scanners 4 – 7 in the hall, scanners 10–12 in the toilets, and scanners 13–19 in the shared bedrooms. Figure 5 shows the route that we walked on this floor. After departing from Node G, we walked slowly for 10 min on the path comprising Nodes G, K, J, C, G, B, A, D, E, I, F, J, K, B, G, C, A, B, K, J, F, I, E, C, G. The true location for each time window was calculated by recording the time at each vertex on the path and by complementation under the assumption that we walked between the nodes at a constant speed. The period of the time window was set to $W_s = 3$ s and the length of the time window was set to $W_w = 6$ s. That is, the two consecutive time windows overlapped by 3 s.

(a)

(b)

Fig. 3. (Color online) BLE tags. (a) Mamorio Fuda. (b) Mamorio on a wristband.

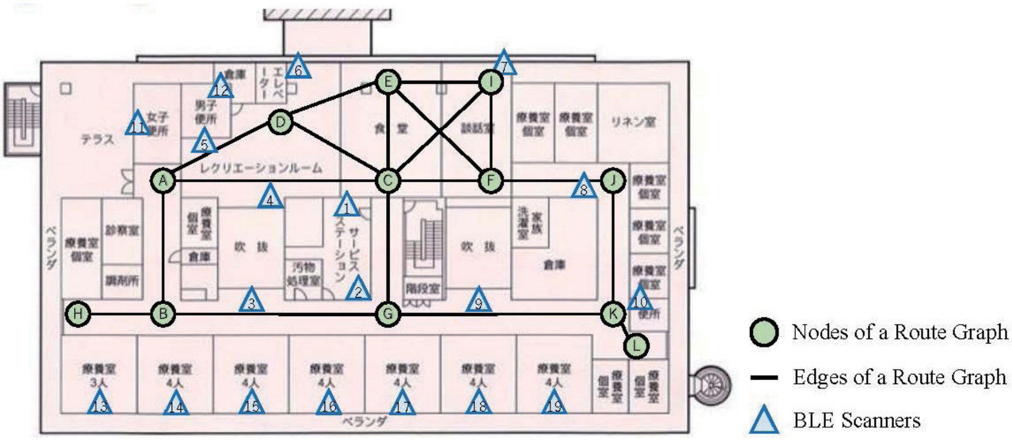

Fig. 4. (Color online) Indoor map with route graph and scanner locations.

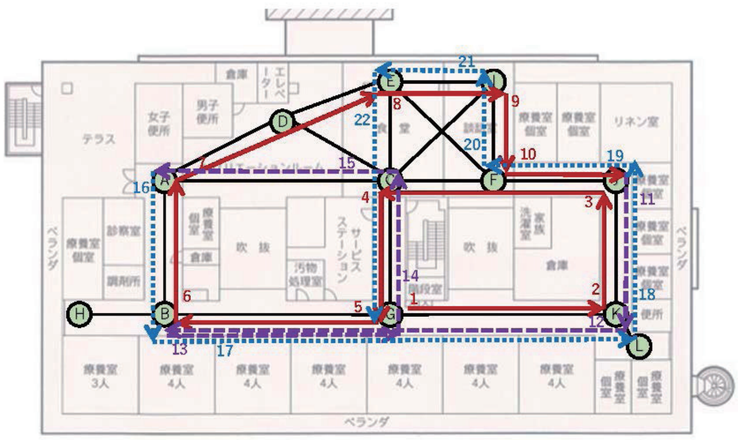

Fig. 5. (Color online) Moving route in evaluation.

We compared the proposed method of GILS with a basic method in which we simply compute the location by least-squares estimation. We compared GILS with this naive method because, to our knowledge, there is no real-time location estimation method with map matching for BLE-based localization. The algorithm of the basic method is as follows. In each time window, we simply compute the most likely location from the distance estimated by the RSSI records. Then we compute the nearest location from it on G within distance from the previous location as the estimated location. If no scanner receives beacons in a time window (remember that we remove the record when three or less beacons are received in the preprocessing), then the estimated location in the time window is missing.

5.2 Error reduction effect with multiple BLE adapters

First, we verified the effect of connecting multiple BLE adapters to a scanner. If the RSSI values of the same BLE beacon are almost the same in all BLE adapters, using multiple BLE adapters does not reduce the location estimation error. Thus, we verified the distribution of RSSI values for each beacon transmission. In the data set obtained in the experiment described above, we identified the RSSI values for each beacon transmission and computed the standard deviation for each beacon transmission when there were four or more RSSI values. Figure 6 shows a histogram of the standard deviation of 67318 beacon transmissions, where the average is 7.10 dB. The result indicates that BLE adapters produce large variations in RSSI values during observation, and thus the use of multiple BLE adapters is effective in reducing errors in distance estimation.

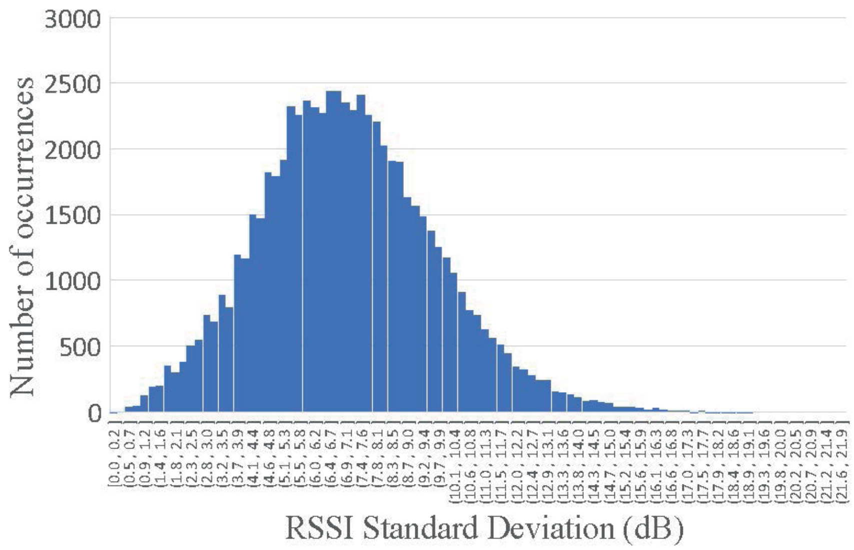

Fig. 6. (Color online) Histogram of RSSI standard deviation for each beacon transmission.

5.3 Evaluating accuracy of position estimation

Next, the accuracy of the position estimation was evaluated. Figure 7 shows the position estimation accuracy of GILS compared with the naive conventional method. The horizontal axis is the elapsed time in our experiment and the vertical axis is the error of each time window. The mean absolute error (MAE) is 3.54 for GILS and 4.10 for the conventional method. The error level is similar in many time windows, but sometimes the conventional method takes a much larger error. This is due to the adjustment using time-series consistency in GILS. Figure 8 shows the number of scanners that observed beacons of the tag in each time window, and Fig. 9 shows the corresponding histogram. The average number of scanners that observed beacons in a time window is 2.74. Less than two (less than three) scanners observed a beacon in the 6 s time window in 28.7% (53.0%) of cases. Since generally three or more scanners are required to estimate the location on the map, this value of 53.0% is rather low for location estimation.

Finally, we found that all moving paths were correctly estimated. Although the number of beacon receptions appears insufficient, the estimated moving paths on G are all correct even when moving through the hall of the map. This indicates that GILS performs well even with sparse scanner placement and with infrequent and low-power beacon transmissions.

5.4 Effect of the time window length

Localization accuracy for various window time W_w is shown in Table 1. Here, we set the window starting interval W_s such that $2W_s = W_w$. The result shows that the value used in the main experiment, $W_w = 6$, is the best. Since the beacon interval is 3 s, it is considered that the number of RSSI values collected in the case $W_w = 3$ is insufficient. Conversely, a larger time window does not improve the accuracy because of the movement within the time window.

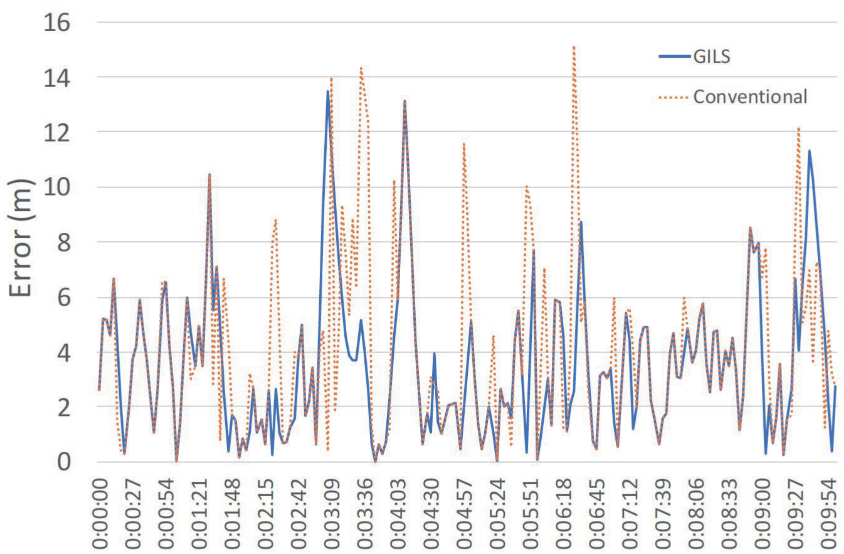

Fig. 7. (Color online) Performance comparison.

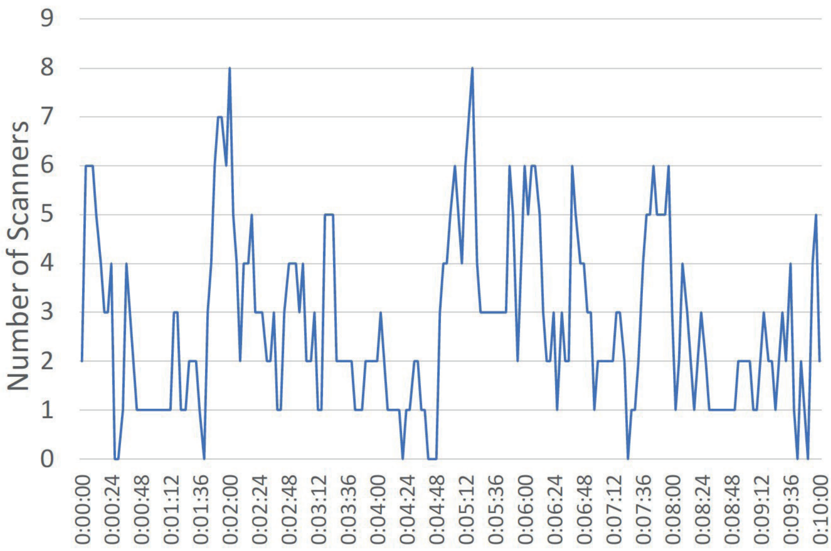

Fig. 8. (Color online) Number of scanners receiving beacons.

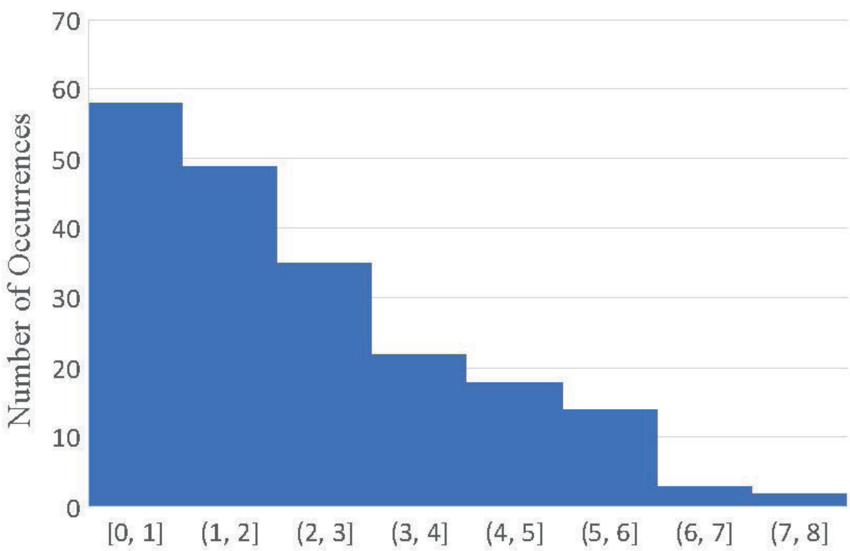

Fig. 9. (Color online) Histogram of scanners receiving beacons.

Table 1
Performance with various window times.

Window time W_w (s)	Mean absolute error (m)
3	3.86
6	3.54
9	3.89
12	3.96

5.5 Limitation of the proposed method

To explore the limitation of the proposed method, we examined the performance with a smaller number of scanners. Figure 10 shows the sparser layouts of the scanners applied. Figure 10 is an 11-scanner layout in which we intend to locate scanners so that each edge is covered by two scanners as much as possible. From this layout, we removed three scanners, 3, 4, and 9, to make an 8-scanner layout, where each node is covered by at least one scanner. We removed scanners 1 and 6 to make a 6-scanner layout. We removed scanner 2 to make a 5-scanner layout.

The result is shown in Table 2, which indicates that the localization accuracy degrades as the number of scanners decreases. The path estimation error also increases as the number of scanners decreases. Note that the path estimation errors in the 11-, 8-, and 6-scanner layouts occur only in the hall on the upper part of the map, e.g., the path A–D–E–I was estimated as A–D–C–I, or the path E–C–G was estimated as E–F–C–G. A reason for this error is that the hall, by nature, is prone to errors in route identification because one can walk anywhere. Another reason will be that scanners 1 and 6 are placed in the corner of the rooms and may have had poor radio reception. Anyway, the practical impact of the path estimation errors in a hall will be relatively small. On the other hand, the path estimation error in the 5–scanner layout occurs in the corridor, e.g., the path B–G–C–A was estimated as B–A. This is because there are no scanners to cover nodes C and G, and this error is not practically acceptable.

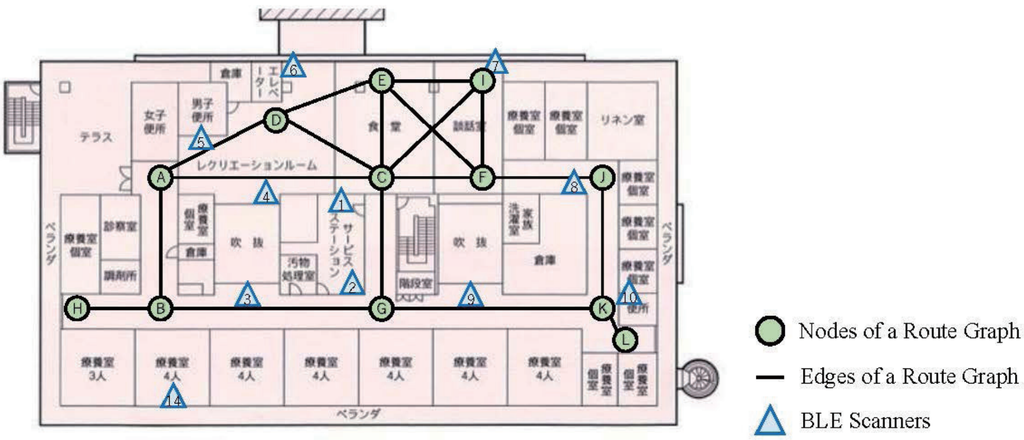

Fig. 10. (Color online) Sparse deployments of BLE scanners (11-scanner layout).

Table 2
Performance with selected scanners.

Number of scanners	Mean absolute error (m)	Correct path rate (%)
20	3.54	100.0
11	4.82	96.2
8	5.36	88.5
6	6.09	80.8
5	7.25	65.4

From the above, we conclude that a sparse scanner layout works if a scanner adequately covers each node of the route graph. It was also suggested that proper placement of the scanner so that the scanner receives good radio reception is also essential to improve the accuracy of position estimation.

6. Conclusion

The size of a beacon tag must be limited for elderly people to wear it continuously, and the time interval and transmission power of beacon transmissions must be limited to reduce battery replacement. In many cases, BLE scanners should be deployed in a sparse arrangement due to installation costs and the location of power outlets, which will reduce the accuracy of location estimation based on RSSI. We presented a solution to this problem by incorporating multiple BLE adapters in a scanner, and we showed that the use of multiple BLE adapters can reduce location estimation errors. We also proposed a real-time location estimation method called GILS that can estimate indoor locations, even when the beacon cycle is long and scanner placement is sparse, by using the route graph on the map and by assuming that the elderly person moves along its edges. We evaluated the performance of GILS using actual devices installed in a nursing home. The results show that GILS provides accurate movement path estimation even under the sparse reception of beacons. A future task is to introduce a statistical model for more robust and accurate indoor localization under sparser situations.

Acknowledgments

We deeply appreciate the cooperation of Mr. Yukihisa Tomari and all the staff members of the nursing home COSUMO. This work was supported by the MEXT Innovation Platform for Society 5.0 Program (Grant Number JPMXP0518071489).

References

1 Statistics Bureau of Japan: https://www.stat.go.jp/data/topics/topi1131.html (accessed March 2022).
2 Cabinet Office of Japan: https://www8.cao.go.jp/kourei/whitepaper/w-2021/html/gaiyou/s1_1.html (accessed March 2022).
3 F. Zafari, A. Gkelias, K. K. Leung: IEEE Commun. Surv. Tutorials **21** (2019) 3. https://doi.org/10.1109/COMST.2019.2911558.
4 R. I. Hartley and P. Sturm: Comput. Vision Image Understand **68** (1997) 2. https://doi.org/10.1006/cviu.1997.0547.
5 K. Urano, K. Hiroi, T. Yonezawa, and N. Kawaguchi: J. Inf. Process. **29** (2021) 58. https://doi.org/10.2197/ipsjjip.29.58.
6 R. Faragher and R. Harle: IEEE J. Selected Areas Commun. **33** (2015) 11. https://doi.org/10.1109/JSAC.2015.2430281.
7 L. Ciabattoni, G. Foresi, A. Monteriu, L. Pepa, D. P. Pagnotta, L. Spalazzi, and F. Verdini: J. Ambient Intell. and Humanized Comput. **10** (2019) 1. https://doi.org/10.1007/s12652-017-0579-0.
8 Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang: Proc. 17th ACM SIGSPATIAL Int Conf. Advances in Geographic Information Systems (2009). https://doi.org/10.1145/1653771.1653820.
9 C. E. White, D. Bernstein, and A. L. Kornhauser: Transp. Res. Part C **8** (2000) 91. https://doi.org/10.1016/S0968-090X(00)00026-7.
10 Z. Xiao, H. Wen, A. Markham, and N. Trigoni: Proc. 13th Int. Symp. Information Processing Sensor Networks (2014). https://doi.org/10.1109/IPSN.2014.6846747.

11 D. Yamamoto, R. Tanaka, S. Kajioka, H. Matsuo, and N. Takahashi: Proc. 26th ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems (2018). https://doi.org/10.1145/3274895.3274918.
12 V. Bloch and M. Pastell: Sensors **20** (2020) 3841. https://doi.org/10.3390/s20143841.
13 J. Krumm and K. Hinckley: Proc. 6th Int. Conf. Ubiquitous Computing (2004). https://doi.org/10.1007/978-3-540-30119-6_17.
14 Mamorio Corporation: https://mamorio.jp (accessed March 2022).
15 H. T. Friis: Proc. I. R. E. **34** (1946) 254. https://doi.org/10.1109/JRPROC.1946.234568.

About the Authors

Takuya Yoshihiro received his B.E., M.I., and Ph.D. degrees from Kyoto University in 1998, 2000, and 2003, respectively. He was an assistant professor at Wakayama University from 2003 to 2009 and has been an associate professor in Wakayama University from 2009. His current interests include graph theory, distributed algorithms, computer networks, wireless networks, medical applications, and bioinformatics. He is a member of IEEE, ACM, IEICE, and IPSJ. (tac@wakayama-u.ac.jp)

Takeshi Goya received his B.E and M.E degrees from Kyoto University, Japan, in 1999 and 2001, respectively. He is currently working as a researcher in Wakayama University. He is interested in IoT system development and web-based applications.