Vol.2017-DPS-171 No.23
Vol.2017-MBL-83 No.23
Vol.2017-ITS-69 No.23
2017/6/2

Location Estimation for Binary Motion Sensors in House

Nathavuth Kitbutrawat[1,a)] Hirozumi Yamaguchi[1,b)] Teruo Higashino[1,c)]

Abstract: Human tracking in household enables estimating the activities and location of residents including children, parents and elderly people. It can be used in the home management system to encourage residents to live in their home comfortable and to reduce the electric expense. Passive infra-red (PIR) motion detection binary sensors are reasonable as they can be portable operated by small batteries and be attached onto walls, shelves, floors or ceilings. However, such flexibility of sensor location causes difficulty since the location information of each sensor should be managed and given to the tracking system by the residents. To mitigate human errors, we propose a sensor localization method to automatically identify the location of multiple binary motion sensors in a house from the observed sequence by binary sensors. The method finds the movement patterns and characteristics of sojourn time of residents to identify the rooms where those sensors are located as well as the proximity relations among sensors, in assistance with floorplan information or room composition obtained by simple questionnaire. The experimental results show that the accuracy of our approach is above 80% after 5 days observation.

1. Introduction

Human tracking and localization systems play an important role for opening a way for serving human need efficiently. In the family house, since there are a variety of ages such as kids, middle-aged and elderly, the indoor human tracking should not require users to wear devices such as smartwatches, which are too invasive for their daily living. Those methods that rely on passive sensors, such as motion sensors for detecting movement of human and contact switches for detecting events when residents open a door [1], [2], are more reasonable solutions to measure the presence and direct motion of those people. In particular, the amount of activities are often required by home healthcare systems for elderly because such systems need to recognize how actively the elderly move even in their houses [3]. However, the existing activity recognition systems need the location of binary sensors to predict the activities of residents by analyzing sequence of events with the Bayesian inference or hidden Markov model. Therefore, the configuration of such passive motion sensors requires a technical labor for the reason that the residents have to configure the coverage area of each sensor, and they might misunderstand the installation location and coverage area. For example, they may put a sensor on the hallway, which should monitor the presence of human in an entrance area. As a result, the system will be unable to recognize human location and activities correctly. Thus the technical employee will be sent to configure the system in customer's home.

There are several drawbacks for sending technicians to configure sensors at home. Firstly, a large number of technicians is needed to deploy the system to many households. Specifically, if a technician spends a few hours to visit a house, configure the system and come back to office, one technician can set up a system in 4 houses per day. Furthermore, for the residents who live in rural places, she/he has to spend longer time for transportation. Secondly, there is always privacy concern. Many residents do not want somebody to come into their houses and survey those places. Therefore, it is desirable that the tracking systems have a self-configuring function to encourage residents to install the system by themselves.

In this technical report, we propose a method to estimate the location of binary motion sensors installed by residents in their houses. This method requests a resident to give some information (*e.g.* floorplan) about her/his house to the system. Then the method analyzes human detection events from those motion sensors to find their locations. It does not require any training procedure before operation. It can find the sensor locations during their operation. This approach requires only the prior knowledge on floor plan analysis to generate an indoor map from a floor plan image. Then we calculate "association rates" between sensors and map them to the installation places in the indoor map. We have conducted two experiments using up to 20 sensors installed to (i) a 2-story house with three family members and

[1] Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
[a)] nat-kit@ist.osaka-u.ac.jpp
[b)] h-yamagu@ist.osaka-u.ac.jp
[c)] higashino@ist.osaka-u.ac.jp

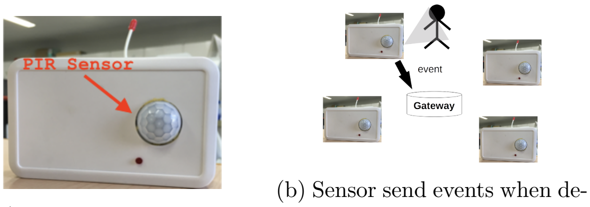
(ii) an apartment with a single elderly. As a result, it could estimate the sensor locations with over 80% accuracy after 5 days operation.

Our main contributions can be summarized into three-hold. Firstly, this is the first approach to localize binary sensors at home. We focus on such labor cost and privacy issue that are incurred by sending technicians to remote houses. Secondly, our approach can find the sensor locations without need of labeled data and supervised techniques. Nevertheless, it can deal with multi-resident environment. Finally, the accuracy has been validated by real-home experiments.

2. Related Work

The previous works on activity recognition based on binary sensors analyze the stream of events from sensors to predict the location of humans and their activities. For example, Hoque et.al. [2] propose a method to detect activities performed during another activity. For example, it may happen that residents suspend to watch television to walk to the toilet and come back to the living room to resume watching television. Thus, they divide activities into short period activities by grouping the sensor events by the time and location of firing sensors, and cluster the set of sensor events which frequently occur together. After that Emi and Stankovic extend the Hoque's work to deal with the multi-resident environment [1]. The idea is to regard the activities occurring in different rooms at the same time as those performed by different users. To sum up, the activity recognition techniques based on binary sensors leverage the location of sensors to predict the location of residents and their activities. Therefore, the system has been configured with the location of sensors before it is operated. To the best of our knowledge, the technique for predicting binary sensor locations in home environment has never been proposed.

Meanwhile, the methods for configuring and calibrating indoor localization systems based on Received Signal Strength (RSSI) have been proposed. In the RSSI-based indoor localization techniques, RSSI is used to measure the distance between wireless sensor nodes and anchor nodes. Then [4] proposes a technique to predict human location by calculating RSSI levels received by Wi-Fi devices from APs with well-known locations. However, for accurate RSSI-based indoor localization, a number of APs should ubiquitously be installed around the building, and the calibration and configuration for such plenty of APs requires numerous effort. To mitigate the cost of calibration and configuration process, there are some researches to configure the location of anchors by requesting a user to carry a Wi-Fi device for collecting the RSSI signals involved in the building. For example, the method by Chintalapudi et.al [5] asks the configuring people to carry a Wi-Fi device for surveying the signal propagation. It measures the distance between devices and Wi-Fi APs by using the signal propagation model and uses the GPS location when the user walks near windows to calculate the actual location of APs. However, those techniques seem unsuitable to our architecture. As discussed earlier,



(a) Our prototype sensor

(b) Sensor send events when detecting human

Fig. 1: PIR Sensor

we do not want to ask residents to perform calibration tasks such as walking along a designated route in a house carrying smart devices, which may cause another problem such as device orientation.

Consequently, this work attempts to find the location of sensors without site survey and data labeling. We will tackle the problem using an unsupervised learning technique. We will describe the details in Section 4.

3. System Architecture

3.1 Overview

In home activity recognition systems, several types of binary sensors are used. Some of them are passive infrared (PIR) sensors, pressure pads and contact switches. This work focuses on finding the location of binary motion sensors (PIR binary sensors) as they are very cost-effective to be penetrating to many households and many products have already been in market. We assume that the motion sensors have to be deployed on wherever residents want to monitor the activities. Since power line supply will be an obstacle for their location-free installation, sensors are built in wireless nodes with low-power communication technology such as ZigBee and are operated by small batteries or energy harvesting as seen in Figure 1a. Those sensor nodes form a wireless sensor network and send events to the gateway when they detect the presence of human shown in Figure 1b.

We attempt to find the location of sensors in multi-resident environment in a general multi-story house. The number of motion sensors which we assume is normally between 10 to 20, but it is not limited to this range. Basically, more sensors will provide finer-grained mobility information, and we will later discuss the impact of the number of sensors to the accuracy. The typical scenario is that firstly a resident obtains (buys or rents) a set of sensors. After the resident deploys those sensors on their own, we request her/him to give some information to our system discussed in Section 3.2. Then we leverage those information to find the location of sensors by analyzing the sequence of sensor events collected through the gateway to the cloud server, without a need in the labeled data discussed in Section 4.

3.2 Indoor Floor Plan

The indoor map is important to provide the information about rooms, hallways, entrance etc. in which residents are able to install motion sensors. The digital versions of given indoor map information can be generated by several techniques. Therefore, we request the residents to take a photo

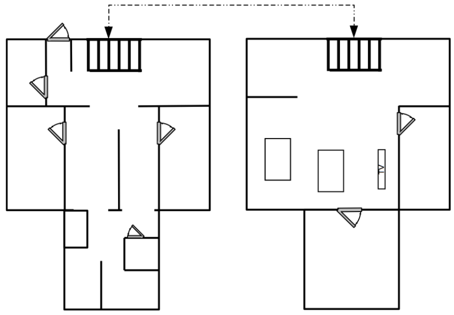


Fig. 2: Floor Plan

of a floor plan image by using a smartphone camera and upload the photo to our system. An example floor plan image is drawn in Figure 2. We note that, some researches provide a solution for large buildings such as office buildings and commercial complex to generate floor plan images by using crowdsourcing techniques. Although the SLAM techniques [6], [7] or PDR techniques [8], [9] to trace human walking paths can be used to construct the floor plan images from walking paths, the drawback is inaccuracy caused by accumulated distance and direction errors caused by orientation change of smartphones during they walk, inaccuracy of stride length estimation and some other unexpected noise.

The given picture of floor plan has to be drawn by well-known symbols such as walls, doors and stairs (such simple illustration can even be hand-drawn). Then we are able to analyze the picture by some existing floor plan analysis technique. For example, the work by Heras et. al. [10] provides a technique to generate the classification model to identify wall, door and window images and to recognize the rooms and space from those identified components. After we analyze the floor plan to detect the areas from the floor plan, we generate a set $L_{floor} = \{l_1, l_2, ..., l_n\}$ of square partitions called *locations*, each of which contains no wall or door. Then we generate a floor plan graph $G_{floor} = (L_{floor}, E_{floor})$ where $(l_i, l_j) \in E_{floor}$ is a walk-through path between two locations l_i and l_j. In particular, we create an edge between two locations if they have a door installed between them or there is no wall between them. We also estimate the area size of each location, and the path distance from the center of l_i to the center of l_j where $(l_i, l_j) \in E_{floor}$ is estimated. We note that a floor plan image is not often accurate to represent the real room structure. For example, a line may be drawn between two locations without any wall between them since lines are sometimes used to represent logical boundaries of two spaces like kitchen and dining spaces. In order to cope with such cases, we tentatively set an edge between two locations if there is some concern that lines may not represent physical walls. If no direct transition occurs between the two locations, the unused edge will be obsoleted in our localization process.

After analyzing the floor plan, we ask through our smartphone App simple questions about the room types. We do not request to associate the sensor ID with the room, which often causes mistakes and troubles. Our question can easily be answered based on their knowledge about their own floor plan. For each highlighted room, they can choose their answer from the list. The listed room types are only limited items, that is, "Common room", "Entrance", "Hallway", "Bedroom", "Kitchen" and "None of the above" to allow easy and quick choices. Through this assistance, we can naturally request the residents to label the rooms.

4. Sensor localization method

We let $S = \{s_1, s_2, ..., s_m\}$ denote the set of motion sensor IDs where m is the total number of sensors. After residents label the locations, the system starts collecting event detection data that are generated by movement of residents at their home from those sensors. Each detection event o_i consists of an sensor ID and a timestamp, and is called *observation*. Hereafter, the sensor ID and timestamp of o_i can be referred to as $o_i.sid$ and $o_i.time$, respectively. We let $O = \{o_1, o_2, ..., o_t\}$ denotes a time sequence of observations called *observation sequence*. Since the detectable area by passive infrared sensors is usually configured to a cone shape with up to a few meters range, each sensor detects humans who pass or stay in a room or a hallway. So the problem can be defined to find a matching function $\mathcal{A} : S \rightarrow L_{floor}$ where S and L_{floor} are sets of sensors IDs and locations respectively, given (i) floor plan graph G_{floor}, (ii) the location type for each location l (denoted as $Type(l)$) and (ii) observation sequence O in a certain period after the sensor placement. As mentioned beforehand, $Type(l)$ is obtained through our smartphone App. We note that we allow multiple sensors being placed in a single location to cover a wide room or for some other reasons.

Our algorithm tries to pick up the best candidate in the following two steps. The first step identifies a set of sensors placed in each "key location" from the observation. A key location corresponds to a typical room like bedroom, kitchen or some others, which usually exists in most of houses. Given observation O, we identify the typical patterns of events that are likely to be seen in such key locations. For example, sensors with events before midnight are probably in the bedroom. Once the locations of those sensors are identified, they act as anchors to identify the locations of the other sensors in the second step. The second step is to find the connectivity between each sensor and estimate the physical distance between them. For this purpose, we create a sensor graph, representing the physical connection of each pair of sensors based on those information. Finally, we choose the best location for each sensor by utilizing the floor plan graph and the sensor graph.

4.1 Identifying Sensors in Key Locations

As introduced in the previous section, we recognize the typical activities that are performed in particular locations (key locations). We use $Type(L_{key} \subset L_{floor}) = \{$"*Kitchen*", "*Bedroom*", "*Entrance*"$\}$ as a set of key locations in this paper. Those sensors installed in the kitchen and bedroom can be identified by analyzing the sensor events that occur within specific time periods, while those at an entrance area can be identified by detecting a "silent time period" (time when residents are out) and focusing on the first and last events in each period that detected the in/out behavior from the house. Concretely, in order to estimate the sensors in the bedroom and kitchen, we would like to know the daily patterns of sleeping and cooking activities over a day. The daily pattern of an activity is the probability for each small period of time in one day, where the residents perform that activity such as cooking and sleeping. Those patterns of general people can be known through crowdsourced survey or some other techniques, and we use the distributions of sleeping and cooking activities, which were obtained by our past survey [11].

We consider the following location-activity pairs {BedRoom,Sleeping} and {Kitchen,Cooking} where the activity is expected to be performed in the location in specific time. The activities such as sleeping and cooking are likely to be noticeable patterns which can be recognized by utilizing the Bayes's theory to analyze the sensor events and the distribution of daily life activity.

We let $Pr(a|t)$ be a probability that people perform activity a in a short time slot t in the daily life. We assume that if a sensor is installed in the kitchen room, it will detect more events than the others while cooking, and a sensor in the bedroom will also do so while (or before and after) sleeping. Therefore, we derive, for each time slot t, the probability that sensor s_i is placed in a location of type $r \in \{Bedroom, Kitchen\}$. This probability is denoted as $Pr(s_i, r|t)$ and defined by Equation 1 below;

$$Pr(s_i, r|t) = \frac{Pr(s_i, r|a, t)Pr(a|t)}{Pr(a|s_i, r)} \quad (1)$$

where $Pr(s_i, r|a, t)$ is the probability that sensor s_i which detected activity a in time slot t is placed in a location of type r. $Pr(a|s_i, r, t)$ is the probability that sensor s_i detects activity a when it is placed in a location of type r. To compute equation $Pr(s_i, r|a, t)$, we define a feature of sensor related with the number of events and time, because the number of events from one sensor varies from time to time, and it may be unsuitable to be analyzed directly. Therefore, we define the feature function $F(s, a, t)$ described below where s, a and t are sensor, activity (cooking or sleeping) and time, respectively.

$$F(s, \text{"cooking"}, t) = \begin{cases} 1 & \text{if } NumE(s, t, 10) \geq \alpha_1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$F(s, \text{"sleeping"}, t) = \begin{cases} 1 & \text{if } NumE(s, t, 60) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In the equation, $NumE(s, t, d)$ returns the number of events from sensor s during the last d minutes from time t, and α_1 is the threshold to detect cooking activity. Empirically, the sensor in a kitchen tends to fire the event over α_1 times in a short time, when a resident cooks a meal, while there are a few events in a bedroom per hour on a sleeping period. Consequently, we use $d = 10$ for cooking activity and $d = 60$ for sleeping activity. The feature value with $\alpha_1 = 10$ and $a =$"cooking" is shown in Figure 3. In this figure, sensors 16, 18, and 19 are placed in the living room, kitchen and dining area, respectively.

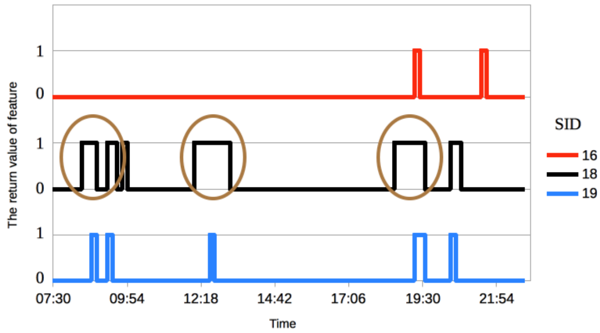


Fig. 3: Value of Feature Function with a = "cooking" (ID=#18)

In the multi-resident environment, it is possible that residents do other activities in other rooms at the same time, and there may be more than one sensor with $F(s, a, t) = 1$. Therefore, we calculate $Pr(s_i, r|a, t)$, by assigning the equal probability to the sensor with $F(s_i, a, t) = 1$ as defined in Equation 4.

$$Pr(s_i, r|a, t) = \frac{F(s_i, a, t)}{\sum_{\forall s_j \in S} F(s_j, a, t)} \quad (4)$$

Finally, $Pr(a|s_i, r, t)$ is calculated by long period data, and we used the time window whose size is up to 7 days in our experiment. During that time window denoted as T_1, we count the number of time slots where $F(s_i, a, t) = 1$ $(\forall t \in T_1)$ considering the prior knowledge $P(a|t)$.

It is given as the following equation (5) where time slot t with non-zero $P(a|t)$ is counted.

$$Pr(a|s_i, r, t) = \frac{\sum_{t \in T_1} F(s_i, a, t) * Pr(a|t)}{\sum_{t \in T_1} F(s_i, a, t)} \quad (5)$$

Then we calculate $Pr(s_i, r)$ which is the average value of $Pr(s_i, r|t)$ over T, and find the maximum value of $Pr(s_i, r)$, $\arg\max_i Pr(s_i, r)$ to predict sensor s_i placed in location of type r.

For sensors in an entrance area, we assume such a duration that all residents are out from home for working or shopping on daytime. During such a period, no events are detected. So we can assume that such sensors installed in entrance area detect the last event before the period, and the first event after the period. We define the human absence vector $ES = \{es_i\}$ where es_i element is the number of periods that sensor s_i detects a leaving house event (2 hours or longer absence) during T. Then we find the maximum value of $Pr(s_i, \text{"entrance"})$ calculated by equation 6 with $\arg\max_i Pr(s_i, \text{"entrance"})$ to predict sensor s_i placed in entrance area.

$$Pr(s_i, \text{"entrance"}) = \frac{es_i}{\sum_{j=1}^{m} es_j} \quad (6)$$

For simplicity of discussion, we assume that there is only

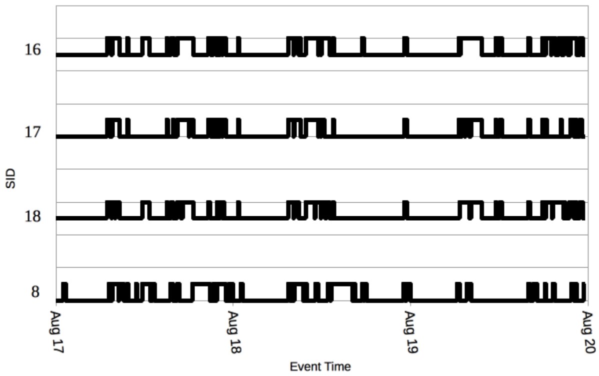


Fig. 4: Similarity of Two Sensor Event Patterns

one sensor for each key location, while the others may have more than one. This is because bedrooms, kitchens and entrances are not so large and one sensor is usually sufficient to cover such a room or space based on our experience of installing sensors to real houses. However, there may be more than one bedroom for a house, that is, more than one location have the same type of locations (*e.g.* a house with two bedrooms). For k locations of the same type r, we may choose those sensors with the top-k probabilities, $P(s_{i_1}, r)$, $P(s_{i_2}, r)$, ..., $P(s_{i_k}, r)$ for key locations of type r.

After obtaining those highest probabilities, we are able to generate the mapping function $\mathcal{A}_{key} : S_{key} \rightarrow L_{key} \subseteq \mathcal{A}$ where S_{key} is a set of estimated sensors in key locations and L_{key} is a set of key locations.

4.2 Sensor Graph

In this section, we estimate the physical distance between each pair of sensor, then generate a sensor graph which represent the predicted walking path between each pair of sensors, and its physical distance based on the event sequence. In particular, if a resident is able to directly walk passing sensor s_i and s_j, the events from those sensors will occur along with each other. Then a traveling time, which is spent for walking passing from s_i to s_j, can be estimated by the physical distance between s_i and s_j. In particular, there is some similarity between two sensors which are placed closely, as exemplified in Figure 4. These are the patterns of sensor events seen in real environment. Sensors 16 and 17 are installed in the same area and their locations are close to the place where sensor 18 is placed, while sensor 8 is far from the others. The patterns of sensor events from sensors 16, 17 and 18 are similar, while the pattern from sensor 8 has less similarity with them.

In summary, the sensors installed in the same room or close location are likely to detect the same movement of human. In particular, when a resident walks out from the area, other sensors in the adjacent area can detect such a walking activity after the previous sensors. In order to estimate the physical distance between each pair of sensors, we focus on the traveling time for walking from one sensor to another. We call a subsequence of O *association event sequence* if it consists of an event sequence from different sensors. For example, for the observation $O = \{o_1, o_2, o_3\}$ where $o_1.sid = s_1$, $o_2.sid = s_1$ and $o_3.sid = s_2$, $\{o_2, o_3\}$ is an association event sequence.

The time duration between the events in an association event sequence basically represents the correlation of those sensors. However, there will be non-confidence time where multi-residents do activities at the same time, and those activities may also create association event sequences between uncorrelated sensors. To deal with this issue, we assume that the traveling time of the same pair of sensors, which is seen when a single resident directly walks passing those two sensors, is relatively constant, while the time intervals caused by multi-residents are dispersed. Therefore, we consider the temporal standard deviation of traveling time $TDIF_{std}(s_u, s_v)$ for each pair of sensors s_u and s_v in T_2 minutes to detect time period during which a single resident is present. Algorithm 1) show the procedure to detect such confident time period.

Algorithm 1 *isConfidence_Time*(O)

Require: $O = (o_0, o_1, ..., o_t)$ is an observation in the past T_2 minutes.
Ensure: *isSingle* which represents the confidence time when it is TRUE.
Ensure: *time_diff* is a set of minimum traveling time between sensors

$t_{start} \leftarrow o_0.time$
for $\forall i \in (1, 2, ..., t)$ **do**
  if $o_i.sid \neq o_{i-1}.sid$ **then**
    if $o_i.sid > o_{i-1}.sid$ **then**
      $s_u \leftarrow o_{i-1}.sid$ and $s_v \leftarrow o_i.sid$
    else
      $s_v \leftarrow o_{i-1}.sid$ and $s_u \leftarrow o_i.sid$
    end if
    $tdif_{s_u,s_v} \leftarrow o_i.time - o_{i-1}.time$
    if $tdif_{s_u,s_v} > 0$ **then**
      update $TDIF_{std}(s_u, s_v)$ by $tdif_{s_u,s_v}$
      $time\_diff_{s_u,s_v} = min(time\_diff_{s_u,s_v}, tdif_{s_u,s_v})$
    end if
  end if
end for
for $\forall e_a \in E_a = (s_u, s_v) \wedge u \neq v$ **do**
  if $TDIF_{std}(s_u, s_v) > \beta_1$ **then**
    $isSingle = FALSE$
  end if
end for
return *isSingle*, *time_diff*

After that, we generate a *sensor graph* $G_{sensor} = (S, E_{sensor})$. This is a fully-connected undirected graph where $E_{sensor} \subseteq S \times S$ is a set of edges representing the physical connection between each pair of sensors. As seen, E_{sensor} basically corresponds to *association event sequence.* Therefore we will remove such an edge $e_{sensor} = (s_u, s_v) \in E_{sensor}$ if there is no *association event sequence* between s_u and s_v. We estimate the traveling time between each pair of sensors by using the minimum traveling time $TDIF_{min}(s_u, s_v)$ to represent the direct walk between each pair of sensor s_u and s_v. The algorithm for generating the sensor graph, $sensor\_graph(O)$ is shown in the algorithm 2.

Algorithm 2 $create\_sensor\_graph(O)$

Require: $O = (o_0, o_1, ..., o_t)$ is an observation in the past T_2 minutes.
Ensure: $G_{sensor} = (S, E_{sensor})$
isSingle, *time_dif* = *isConfidence_Time*(O)
create $G_{sensor} = (S, (S \times S))$
if *isSingle* =TRUE **then**
  for $\forall s_u, \forall s_v \in S$ $(s_u \neq s_v)$ **do**
    update $TDIF_{min}(s_u, s_v)$ by $time\_dif_{s_u,s_v}$
    if $TDIF_{min}(s_u, s_v) = 0$ **then**
      remove edge (s_u, s_v)
    else
      $(s_u, s_v).interval \leftarrow TDIF_{min}(s_u, s_v)$
    end if
  end for
end if
$walking\_speed \leftarrow \beta_2$
for $e \in E_{sensor}$ **do**
  $e.distance \leftarrow e.interval * walking\_speed$
end for

In our experiment, the constant β_2 is set to 1 m/s to calculate the estimated physical distance between each pair of sensors. Then we will use the sensor graph to map each sensor $s \in S$ into location $l \in L$ leveraging the floor plan graph and the sensor graph.

4.3 Matching

In this section, we create the mapping function $\mathcal{A} : S \rightarrow L_{floor}$ where L_{floor} is a set of room locations from the floor plan graph, and S is a set of sensors in the sensor graph. We leverage the room dimensions and physical distance between one room and its adjacent rooms in the floor plan graph along with the physical distance in the sensor graph to estimate the locations of sensors.

We introduce an sensor-room association score $assoc_{s,l}$ which represents the number of times where sensor s is expected to be placed in that location l. The process to increase the $assoc_{s,l}$ and select the sensor location is shown in the following two steps.

In the first step, we start by the sensors in S_{key} because the locations of those sensors are likely to be correctly estimated. Then we classify the rest of sensors which have edges to those in the key locations in the sensor graph into 3 groups: "same location", "next location" and "unknown". In particular, sensor s_i will be placed in the same location (room) with sensor $s_{key} \in S_{key}$ if a physical distance from s_i to s_{key} is lower than that room dimensions. Sensor s_i is estimated to be placed in location l_i which is next to location l_{key} of sensor s_{key} if a physical distance from s_i to s_{key} is lower than the summation of the distance between l_i and l_j and that room l_i dimensions. Otherwise sensor s_i is classified into the "unknown" category. Then we increase $assoc_{s_i,l_{key}}$ and $assoc_{s_i,l_i}$ by 1 when sensor s_i is labeled as "same location" and "next location", respectively.

In the second step, we find $\forall s \in S; \arg\max_l \ assoc_{s,l}$. This means the location with the highest score is regarded as the location of s. Then we will update the mapping function $\mathcal{A}$ by appending "$\mathcal{A}(s) = l$". After that our algorithm repeats the step one until the algorithm is unable to find the maximum score in $score_{s,l}$ shown in algorithm 3.

Algorithm 3 $matching(G_{floor}, G_{sensor}, \mathcal{A}_{key})$

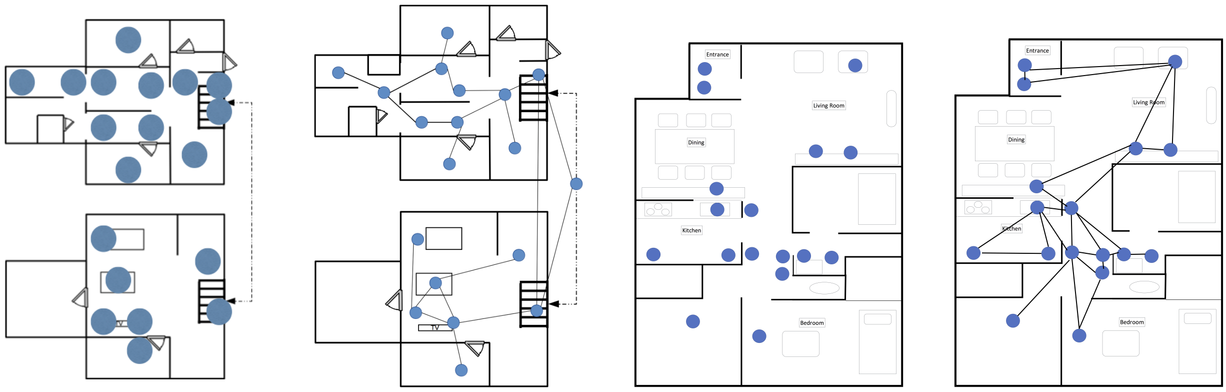
Require: $G_{floor} = (L_{floor}, E_{floor})$ is a floor plan graph. $E \subseteq L_{floor} \times L_{floor}$ contains some attributes such as physical distance between two locations.
Require: $G_{sensor} = (S, E_{sensor})$ is a sensor graph. $E_{sensor} \subseteq S \times S$ contains some attributes such as physical distance between two sensors.
Ensure: $\mathcal{A} : S \rightarrow L_{floor}$
satisfy = FALSE
while $\neg$*satify* **do**
  for $\forall e = (s_i, s_j) \in E_{sensor}$ where $s_i \in S_{key}$ **do**
    $l = \mathcal{A}(s_i)$
    $roomsize = min(l.width, l.heigh)$
    if $e.distance \leq roomsize$ **then**
      $assoc_{s_j,l} = assoc_{s_j,l} + 1$
    else
      N_l as set of neighbor locations of l in G_{floor}
      for $\forall e' = (l, l') \in E_{floor}$ where $l' \in N_l$ **do**
        if $e'.distance \leq e'.distance + roomsize$ **then**
          $assoc_{s_j,l'} = assoc_{s_j,l'} + 1$
        end if
      end for
    end if
  end for
  satify =TRUE
  for $\forall s \in S \wedge s \notin S_{key}$ **do**
    $L_{list} = \arg\max_l \ assoc_{s,l}$
    if $|L_{list}| == 1$ **then**
      $\mathcal{A}_{key}(s) = head(L_{list})$
      satify =FALSE
    end if
  end for
end while
return $\mathcal{A}_{key}$

5. Experiment

5.1 Dataset and Scenario

We conducted the experiments in two real houses. Firstly, sensors were installed in a real two-story house with three family members. The first dataset called DT1 was collected from 19 motion sensors in this two-story house (approximately 120 square meters) in mid of 2016 (shown in Figure 5a). The second dataset called DT2 was collected from 18 motion sensors in the same house in early 2015. Next, we installed 18 motion sensors to the single story house (approximately 40 square meters) where a single resident (elderly person) lives. The last dataset called DT3 was collected from the house in February 2017 (shown in Figure 5c and Figure 6). We assume that the floor plans of these houses have already been identified before location estimation.

After our system collected events for one week, we created



(a) Sensor Location in DT1 (b) Sensor Graph in DT1-C1 (c) Sensor Location in DT3 (d) Sensor Graph in DT3-C1

Fig. 5: Floor Plan with Sensor Locations

Fig. 6: Snapshot from Experiment Site

3 scenarios from each dataset. The first scenario C1 is to use the original data in the dataset. The second scenario C2 is to use only one sensor per one area, where the sensor events from some sensors installed in the same area are removed. The last scenario C3 considers the privacy concerns where the event data from sensors in bedrooms are hidden. We generated G_{sensor} seen in Figures 5b and 5d from DT1-C1 and DT3-C3 scenarios, respectively. The evaluation was done using our python program built for this purpose.

5.2 Definition of Feasible Solutions

In the evaluation, we have introduced some acceptable cases when we verify our result since there are some possible solutions of matching in symmetric structure. Although our system uses only the motion sensors detecting the movement of residents, we are unable to identify locations depending on the symmetry as seen in Figure 7.

In Figure 7a, let us suppose sensor k is in a key location, and sensors 2 and 4 in the top-side rooms are possible to switch the positions with sensors 1 and 3 in the bottom-side rooms (and similarly, there is left-right case; (1,2) and (3,4)). Therefore, we define 2 rules that regard both cases as feasible solutions; (i) inner-room switching (the top-bottom example) and (ii) inter-room switching (left-right example). Figures 7b and 7c show more clearly the acceptable and unacceptable cases.

Table 1: Matching Accuracy

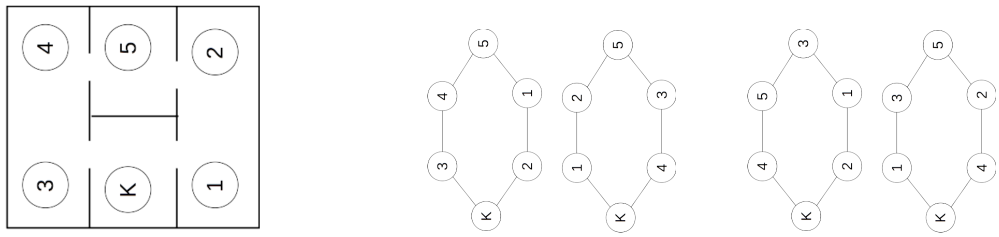
	Running Days					
Scenario	1	2	3	4	5	6
DT1-C1(19 sensors)	0.16	0.58	0.68	0.68	0.89	0.89
DT1-C2(12 sensors)	0.17	0.75	0.83	0.67	0.83	0.83
DT1-C3(17 sensors)	0.18	0.53	0.53	0.41	0.59	0.65
DT2-C1(18 sensors)	0.29	0.35	0.59	0.71	0.82	0.88
DT2-C2(11 sensors)	0.27	0.27	0.73	0.82	0.82	0.82
DT2-C3(16 sensors)	0.20	0.33	0.33	0.47	0.67	0.70
DT3-C1(18 sensors)	0.33	0.47	0.67	0.67	0.73	0.75
DT3-C2(7 sensors)	0.57	0.71	0.71	0.71	0.71	0.86
DT3-C3(17 sensors)	0.36	0.50	0.43	0.57	0.64	0.64

5.3 Matching Performance

We run the matching algorithm one time when we have collected for a day, because some daily life activities are usually done only one time in one day period. Especially, everyone sleeps once a day in normal situation. Then we found the accuracy of results in 4 scenarios increases to over 80% within 5 days as seen in Table 1.

In the result, we found the good accuracy of location estimation in scenarios DT1-C1, DT1-C2, DT2-C1, DT2-C2 and DT3-C2, because the sufficient number of sensors are used in those scenarios. Especially, the result in scenario DT3-C2 had a good accuracy after collecting data for 2 days, because there is only a single resident living in the house and our algorithm can identify the physical connection between each pair of sensors easily. Meanwhile, the accuracy in the scenario DT3-C1 is low despite there are many sensors deployed. The reason is when the abundant sensors are deployed, most of them may monitor the same regions, and make the structure of sensor graph complex. This causes some difficulty in identifying their locations correctly. We note that we found the invalid prediction for sensors in key locations when data have been collected only for 1 day or 2

Vol.2017-DPS-171 No.23
Vol.2017-MBL-83 No.23
Vol.2017-ITS-69 No.23
2017/6/2



(a) Example: sensors in symmetric structure (b) Acceptable result (c) Unacceptable result

Fig. 7: Symmetric Structure

days. The lowest accuracy is the case of scenario DT1-C3, DT2-C3 and DT3-C3. In those cases, we lost key location information (about bedroom; remember C3 is such a scenario) and the bedroom looked like outside (since no sensor was there). But finally, it is recognized, based on the patterns $P(a|t)$, the sensor in front of the bedroom was regarded as non-entrance and finally the accuracy was improved as time passed. In any case, after 7 days' data collection, the accuracy was over 70%.

6. Discussion

In this paper, we show our algorithm can estimate the sensor locations, which is a room-level localization of sensors. We found that there are two main obstacles which hinder the increase of accuracy in the localization. Firstly, the symmetric structure in the real house hinders our techniques to estimate the location of sensors correctly. The second obstacle is it is sometimes difficult to estimate the appropriate number of sensors that is enough for estimating their locations with high accuracy. Although the more we deploy sensors in a house, the more the accuracy in activity recognition will increase. This has been proved in the C3 scenario. We will conduct more experiments to observe how the algorithm works in different situations.

7. Conclusion

This paper proposed a method to find the location of motion sensors that are used to detect the daily life activities in home environment. Our technique requires only on some prior knowledge such as floor plan, and it is able to find the mapping function between sensors and locations over 80% accuracy after 5 days data collection without annotation. Our future works includes more experiments in real and simulated environments, assuming a variety of house types in terms of area sizes, the number of rooms, the number of residents and the characteristics of residents (age, gender, family composition etc).

Acknowledgment

This work was supported in part by JSPS KAKENHI JP15H02690, JP15K12019 and JP2622000.

References

[1] I. A. Emi and J. A. Stankovic, "Sarrima: Smart adl recognizer and resident identifier in multi-resident accommodations," in *Proceedings of the Conference on Wireless Health*, ser. WH '15. New York, NY, USA: ACM, 2015, pp. 4:1–4:8.

[2] E. Hoque and J. Stankovic, "Aalo: Activity recognition in smart homes using active learning in the presence of overlapped activities," in *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, May 2012, pp. 139–146.

[3] Harvard Health Publications, 2016. [Online]. Available: http://www.health.harvard.edu/diet-and-weight-loss/calories-burned-in-30-minutes-of-leisure-and-routine-activities

[4] C. Morelli, M. Nicoli, V. Rampa, U. Spagnolini, and C. Alippi, "Particle filters for rss-based localization in wireless sensor networks: An experimental study," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 4, May 2006, pp. IV–IV.

[5] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '10. New York, NY, USA: ACM, 2010, pp. 173–184.

[6] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 889–898, Nov 2012.

[7] M. Hardegger, D. Roggen, S. Mazilu, and G. Tröster, "Actionslam: Using location-related actions as landmarks in pedestrian slam," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nov 2012, pp. 1–10.

[8] D. Philipp, P. Baier, C. Dibak, F. Dürr, K. Rothermel, S. Becker, M. Peter, and D. Fritsch, "Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2014, pp. 139–147.

[9] M. Alzantot and M. Youssef, "Crowdinside: Automatic construction of indoor floorplans," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '12. New York, NY, USA: ACM, 2012, pp. 99–108.

[10] L.-P. de las Heras, S. Ahmed, M. Liwicki, E. Valveny, and G. Sánchez, "Statistical segmentation and structural recognition for floor plan interpretation," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 3, pp. 221–237, 2014.

[11] S. Nakamura, S. Shigaki, A. Hiromori, H. Yamaguchi, and T. Higashino, "A model-based approach to support smart and social home living," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '15. New York, NY, USA: ACM, 2015, pp. 1101–1105. [Online]. Available: http://doi.acm.org/10.1145/2750858.2805835