M-034

Location estimation of PIR sensor for sparse deployment

Nathavuth Kitbutrawat*, François Portet*†, Hirozumi Yamaguchi* and Teruo Higashino*
Graduate School of Information Science and Technology, Osaka University, Japan*,
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France†

1. Introduction

Localization service in a household is one of the critical technologies to provide a suitable service for the resident based on his or her location in houses. Since residents do not carry constantly a device when they live in their houses, the techniques which require resident to be along with a device are unusable [1]. Hence, passive motion sensors such as Passive Infra-Red (PIR) sensors, which can detect the movement of humans without a need of carried device, are the industrial standard to provide such a service. Furthermore, there is a rising number of research studies showing that PIR can be exploited to perform human tracking [2], [3]. However, there are some disadvantages in terms of deployment cost and privacy concern to send a technical team to households. End-users might also be insufficiently experienced to install and manage a network of PIR sensors, especially for typical residents, such as older adults, with low technical skill. Therefore, to enable easy deployment of sensor network by anybody in their home, it is desirable that the home automation system has a self-configuring function so that residents can install the systems by themselves easily.

In this paper, we propose a method that automatically associate PIR sensors to their room location with a low amount of a priori knowledge. Such localization method would be the core building block of a self-configuring system which would help end-user deploying sensors effectively.

2. Methodology

To identify location of PIR sensor, we assume the typical scenario is that residents deploy the system on their own. Firstly, they obtain (buy or rent) a set of sensors. After the resident deploys those sensors on their own, they are requested to upload a floor plan image to a server. At the same time, sensors start detecting movements of residents and send the sensor events to the server. Finally, the sensor events are analyzed and the location of every sensor is estimated. By Estimating the travel time between pairs of sensors, we are able to perform a graph matching, and seek the best matching function to map sensor into the location by utilizing a distance between rooms (from floorplan) and the travelling time (from sensor events). The specific detail is described below.

2.1 Floor plan Graph

We introduce a floor plan graph G_{floor} models the possible pathway (edge) from one location (node) to another location. To generate such graph, a floor plan image is provided to extract a set $L_{floor} = \{l_1, l_2, ..., l_n\}$ of rooms, the distance function $f_d : E_{floor} \rightarrow R^+$, and a matching function $f_t : L_{floor} \rightarrow T$ to match a room $l_i \in L_{floor}$ to its room type where T = {entrance, corridor, kitchen, bathroom, living room, bedroom}. Then, the floor plan graph $G_{floor} = (L_{floor}, E_{floor}, f_t, f_d)$ is generated where an edge in E_{floor} represents two locations a human can directly walk from one to another. In this paper, we assume that the floor plan graph has already identified by prior knowledge.

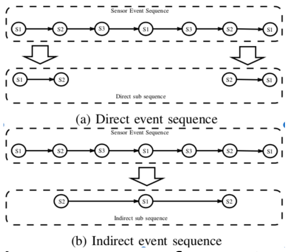

Fig. 1 Subsequence of event sequence

2.2 Sensor Graph

To estimate the location of sensors, the method first con- structs a sensor graph $G_{sensor} = (S, E_{sensor}, T_{sensor})$, where S is a set of sensors deployed in a house, an edge $e_{sensor} = (s_i, s_j) \in E_{sensor}$ represents the fact that there was a direct trip between sensor s_i and sensor s_j, and $t_{si,sj} \in T_{sensor}$ is an estimated trip time when a resident walks from sensor s_i to sensor s_j. This sensor graph is estimated from a sequence of events seq_{event} which is a time sequence of motion detection events.

In literature [4], sensors are deployed densely, thus we can generate the E_{sensor} by considering the number of event sensors from each pair of the sensor. However, the resident can pass the sensor in the middle in the scattered case. As a result, we adjust the method to deal with this case. We, therefore, introduce the notion of ***direct event sequence*** of pair (s_i, s_j), which is a sub sequence of events between sensor s_i and sensor s_j having no event from another sensor in the middle (i.e., the direct successor) as in Fig. 1 (a). On the contrary, an ***indirect event sequence*** of (s_i, s_j) is a sub sequence of event sequence between sensors s_i and s_j having event(s) from other sensor(s) in between (i.e., not direct successor) as in Fig. 1 (b).

The existence of an edge $e_{sensor} = (s_i, s_j) \in E_{sensor}$ between sensor s_i and sensor s_j is estimated by considering the number of direct event sequence of pair (s_i, s_j) using a threshold. Finally, the trip time $t_{si,sj} \in T_{sensor}$ that the resident spends on walking on every edge e_{sensor} can be estimated. Briefly, our method analyzes both direct and indirect event sequences of sensor pairs (s_i, s_j). For example, we analyze an event sequence $seq_{event} = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), (s_1, t_4), (s_3, t_5), (s_2, t_6), (s_1, t_7)\}$ and generate a set of trip times $X_{s1,s2} = \{t_2 - t_1, t_4 - t_2, t_6 - t_4, t_7 - t_6\}$ of pair (s_1, s_2). After generating $X_{si,sj}$, the method has two steps.

2.2.1 Clustering

The clustering step models different trip patterns (fast moving, typical trip, stop-and-go travel between loca- tion) from $X_{s1,s2}$ in order to filter out irrelevant times by utilizing GMM clustering.

2.2.2 The estimation

The estimation step computes the trip time from the remaining clusters. Specifically, the median of the trip time instances belonging to the two first clusters (shortest trip time values) is supposed to be the estimated trip time $t_{si,sj} \in T_{sensor}$. As a result, the clusters with the longest trip time values are assumed to due to stop-and-go behaviours or long displacements across the home and we can eliminate them.

2.3 Matching

In this study, we examine every matching pattern. For example, if we deploy d sensors in a house which contains $_n$ rooms (d is smaller than n) and if we identify k sensors in the key location, we have to consider $_nP_{d-k}$ matching patterns. In each pattern, we compare the estimated trip time and the distance between rooms by a slack value Δ_1 to consider the possibility of this pattern. For example, we consider the matching function $A : S \rightarrow L_{floor}$, our method calculates the shortest distance from location l_u to location l_v in floor plan graph G_{floor}. After that, we estimate the trip time $t_{lu,lv}$ from location l_u to location l_v by using the velocity based on the recent studies about gait [5].

We denote $A_{all} = \{A_1 , A_2 , ...\}$ is a set of all matching pattern where A_1,A_2 are the matching function $A : S \rightarrow L_{floor}$. Since, the number of rooms is greater than the number of sensors in scattered deployment. We, thus, introduce a feasible location set for each sensor, which is a set of locations being able to match to a sensor. Normally, every location is a member in feasible lists except the key location and the feasible list for a sensor in key location has one member. After that we apply the linear programming to seek all possible matching function.

For each possible matching A where $A{:}s_i \rightarrow l_u$ and $A{:}s_j \rightarrow l_v$, we calculate the matching score $score(A)$ by using following equation.

$$score(A) = \sum_{\forall s_i, s_j} \sum_{\forall l_u, l_v} (|t_{s_i,s_j} - t_{l_u,l_v}|)$$

We rank the matching score $score(A)$ and select the matching patterns whose the $score(A)$ is lower than a slack value Δ_1. In this work, the slack value Δ_1 is calculated by the lowest matching score plus 10%. Then we generate the matching frequency matrix M (m_{iu}) which represent how many time that sensor s_i is matched to location l_u, and we match the sensor $s_i \in S$ to the location $l \in L_{floor}$ when the m_{iu} is highest for each m_{i*}.

3. Experiment

3.1 Dataset

To evaluate the approach on set of scattered Infrared sensors, we used the ContextAct@A4H dataset [6]. It is a rich, real-life daily living dataset collected in the Amiqual4Home smarthome. This Smart Home is fully functional and equipped with more that 500 controllable or observable items (e. g., lighting, shutters, security systems, energy management, heating, etc.). Among the sensors, six PIR binary sensors were set in the ceiling of the kitchen, the living room, above the dinner table, above the bed of the bedroom, in the office and bathroom.

The ContextAct@A4H dataset was collected while a person was living there alone during 30 days in June and November (summer and fall respectively). This collection resulted in 30756 PIR firing.

3.2 Matching performance

We analyze the matching between sensors and room locations. The raw accuracy of matching result is 55%. However, when confusions of couch and table are reconsidered as true positive since they cover similar room, the accuracy becomes 78%. Highly confused rooms are bathroom and office. This is due to the fact they are very close in distance from the bedroom which was the key location. Hence a symmetry in the graph of sensors appeared and the two sensors were difficult to distinguish. If a fourth sensors was present in floor, this kind of symmetry could have been resolved. Similar problem can be emphasized for the confusion between table, entrance, toilets and stairs which are at similar distance from the kitchen.

4. Conclusion

In this paper, we have presented the results of a method to estimate automatically the location of a set of PIR binary sensors in home. The method is unsupervised and just needs the floor plan of the house and one or two key locations as prior knowledge. The results of the study show the difficulty of the task with a location accuracy from 55% to 78%. In particular, the small set of sensors decreases the redundancy in the values and hence the robustness of distance computation.

Acknowledgement

The work was supported by ”Research and development of Innovative Network Technologies to Create the Future (No. 191)” ,the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.

Reference

[1] N. C. Krishnan and D. J. Cook, “Activity recognition on streaming sensor data,” Pervasive and Mobile Computing, vol. 10, pp. 138 – 154, 2014.
[2] A. S. Crandall and D. J. Cook, “Tracking systems for multiple smart home residents,” in Human behavior recognition technologies: Intelligent applications for monitoring and security, 2013, pp. 111–129.
[3] T. Miyazaki and Y. Kasama, “Multiple human tracking using binary infrared sensors,” Sensors, vol. 15, no. 6, pp. 13 459–13 476, 2015.
[4] N. Kitbutrawat, H. Yamaguchi, and T. Higashino, “Localization of binary motion sensors in house,” in 13th International Wireless Communica- tions and Mobile Computing Conference, IWCMC 2017, Valencia, Spain, June 26-30, 2017, 2017, pp. 1132–1137.
[5] M.Schimpl,C.Moore,C.Lederer,A.Neuhaus,J.Sambrook,J.Danesh, W. Ouwehand, and M. Daumer, “Association between walking speed and age in healthy, free-living individuals using mobile accelerometry-a cross-sectional study,” PLoS ONE, vol. 6, no. 8, 2011.
[6] P. Lago, F. Lang, C. Roncancio, C. Jime ́nez-Guar ́ın, R. Mateescu, and N. Bonnefond, “The ContextAct@A4H real-life dataset of daily-living activities Activity recognition using model checking,” in CONTEXT, ser. LNCS, vol. 10257, 2017, pp. 175–188.