Automatic Localization of Passive Infra-Red Binary Sensors in Home: from Dense to Scattered Network

Nathavuth Kitbutrawat, François Portet, Hirozumi Yamaguchi, Teruo Higashino

Automatic Localization of Passive Infra-Red Binary Sensors in Home: from Dense to Scattered Network

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†
Email: nat-kit@ist.osaka-u.ac.jp, francois.portet@imag.fr, h-yamagu@ist.osaka-u.ac.jp, higashino@ist.osaka-u.ac.jp

***Abstract*—Location of residents in a household is one of the critical information to provide context-aware services. Passive Infra-Red (PIR) binary motion sensors have become the *de facto* standard technology used in the home by tracking systems due to their low energy consumption and their wide range of coverage. However, installing and managing this network of PIR sensors is difficult for typical residents, such as older adults, with low technical skill. To enable easy deployment of such a system by anybody, we present an extension of a method to automatically identify the location of multiple PIR sensors in a house from the observed motion detection event sequences. Thanks to a floor plan given as prior knowledge, the method estimates the distance between pairs of sensors and identifies particular patterns in the observations to predict the rooms where those sensors are most likely located. The method, which was designed to deal with dense sensors network is adapted to the case of scattered sensors which correspond to most traditional houses. Experimental results on a realistic home show that our method can estimate the location of sensors placed close to the anchor locations with only a few confusions. The experiments also revealed challenges to be addressed to make this method scale to various house configurations.**

I. 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. For instance, a light and air conditioner could be programmed to be turned on when the resident enters a room. Indoor localization techniques which require users to wear devices such as smart-watches or to carry a smartphone are not adequate solutions since residents do not carry constantly a smart device when they live in their houses [1], [2], [3], [4]. Hence, passive motion sensors such as Passive Infra-Red (PIR) sensors, which can detect the movement of humans, 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 [5], [6].

Hence, PIR sensors combine the advantages of being free from carried devices, being accurate, and being available on the market. However, to provide a location service efficiently, setting such a set of PIR sensors requires technical workers to be sent to the household to deploy the system. Indeed, technicians must connect the PIR sensors to the house network and configure the exact location of them to deliver efficient services. 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. Our localization method brings the following advantages:

- Distances between sensors are automatically computed from the sensors readings in a unsupervised manner.
- Sensors are mapped to rooms using a floor plan given by the user. Hence the locations to which sensors are mapped are directly understandable by the user. The automatic extraction of house-related spaces from the floor plan is out of the scope of this paper. It suffices to stress that recent research work has shown that automatic analysis of floor plan is feasible [7].
- The method can exploit a variable amount of *a priori* knowledge. The user can specify one to several sensor (key) locations to speed-up the location process.
- The method is designed to work with the common cases of scattered PIR network where not all rooms are equipped with sensors. Yet, it can also address dense network cases (fully equipped smart home, offices) [8], [9].

The paper is structured as follow. After a short overview of the related work in Section II, the method is described in detail in Section III. Briefly, to perform localization of binary motion sensors in a house where a small number of binary sensors are deployed, similarly to [8], [9], the resident is requested to provide the floor-plan of her/his house to the system as well as the *key location* of one or two sensors. Once the floor plan and the sensors at the key location are identified, the remaining sensors of the network are automatically localized by analyzing the binary sensor events to estimate the trip time between two sensors where a resident walks directly between them. Then, a room-level mapping is performed to associate the sensors to the location in the floor-plan. The method has

been applied to realistic data and both the dataset and the results of the experiment are presented Section IV. The results show that, despite the low amount of information present in a scattered network or PIR, the method is able to identify the most probable room location of the sensors. The paper ends with a short conclusion and emphasizes the challenges to be addressed in further work.

II. Related work

Contrary to human tracking and human activity recognition, automatic unsupervised localization of sensors in a home is an under-explored research area in the pervasive community. In the real house, the passive motion sensor-based localization is a suitable technique to be deployed due to its cost and device-free manner. Since we cannot be applied to the passive motion sensor-based localization because the sensors are configured as client mode (receiver mode) due to battery limitation. Hence the existing sensor localization such as WiFi-based localization of sensor e.g. Path attenuation [10] and Time of Arrival [11] cannot be used.

Closely related work is the one by Twomey *et al.* [12] which extracts an adjacency matrix of sensors in an unsupervised manner to determine which sets of sensors are the more adequate to use to improve human activity recognition. However, the adjacency matrix was not built to associate the sensors to specific rooms. The closest work is the one of Kitbutrawat *et al.*[8], [9], within which a sensor localization system from PIR sensors data was investigated. The method automatically identified the location of multiple binary motion sensors (about 18 to 21) in a house from the sequence of binary sensors observations. The method maps the PIR observations to rooms in a pre-defined floor-plan. The method exhibited an accuracy of above 80%. However, it requires a dense network of short-range homemade PIR sensors. This is not the current situation of a standard home which contains a low amount of long-range PIR sensors positioned specifically for lighting and alarm. Going from dense networks to scattered ones implies a sharp decrease of redundancy in the measurements, and the larger area of firing decreases the location accuracy.

III. Methodology

In this study, we focus on a method to identify the location of passive motion sensors since those sensors can detect the presence of a resident without the need of explicit human intervention. We assume that the passive motion sensors considered in this study fire an event when they detect the presence of humans, but they do not fire an event when the presence of resident disappears for energy saving.

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.

A. Problem Definition and Assumptions

In order to perform this localization, two tasks must be performed:

- Estimating the travel time between pairs of sensors
- Mapping sensors to room (or sub-room) location in a floor plan

The time difference between two sensor readings depends on several factors: the real distance of sensors $dist$ (the value we want to find), the firing delay (e.g. PIR) δ, the travel velocity ν, the PIR range detection (e.g., PIR reception cone) a, the measurement noise ϵ and the interference between several dwellers ζ. Hence, Consequently the time difference between two consecutive observations could be seen as :

$$|t_i - t_j|_{i \neq j} = \begin{cases} \frac{(dist(s_i,s_j)-a)}{\nu} - \delta + \epsilon & \text{when one dweller is assumed} \\ \zeta & \text{otherwise} \end{cases} \quad (1)$$

However, it is unknown which dweller related to each observation and ν is not directly observable leading to a solution that can only be approximated. In this study, we consider observable variables as independent random variables contributing to the distribution of $|t_i - t_j|$. Furthermore we consider the following assumptions:

- in the unique dweller case, low values of $|t_i - t_j|_{i \neq j}$ are due to highest ν that are not stop-and-go travel
- in the multi-dweller case, lowest values of $|t_i - t_j|_{i \neq j}$ can be due to ζ
- high values of $|t_i - t_j|_{i \neq j}$ are due to either ζ or low ν (stop and go behaviour)

Thus restricting the study to one-dweller case permits to ignore the ζ case. Furthermore in the remaining of the paper we assume a and δ to be constant and ϵ small enough to be ignored. Hence, in summary, it is assumed that $|t_i - t_j|_{i \neq j}$ mainly depends on ν. Thus, our strategy is to compute the distribution of $|t_i - t_j|_{i \neq j}$ on sufficient amount of samples and find the samples that are the most likely to belong to the highest velocity values. Once these samples are selected, they can be used to compute the approximation of $dist(s_i, s_j)$ for each pair of sensors. A *sensor graph* G_{sensor} is obtained.

Then, once the set of distances between pairs of sensors is estimated, the mapping to room location is performed assuming that one or two key locations of sensors are given. Then, each remaining sensor is mapped incrementally to a *floor plan graph* G_{floor} introduced below, following a greedy search.

In the greedy search, we perform a graph matching, and seek the best matching function by utilizing a distance between rooms and the travelling time where resident spend on walking on that path. For specific detail, we explain in section III-D.

B. floor plan graph

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 walking distance between two rooms, and a matching function $f_t : L_{floor} \rightarrow T$ to match a room $l_i \in L_{floor}$ to its room type $t_(l, i) \in T$ where $T = \{entrance, corridor, kitchen, bathroom, living\_room, bedroom\}$[1]. 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, and the distance function $f_d : E_{floor} \rightarrow R^+$ gives the Euclidean distance between the centers of l_i and l_j for each $(l_i, l_j) \in E_{floor}$.

C. Sensor Graph

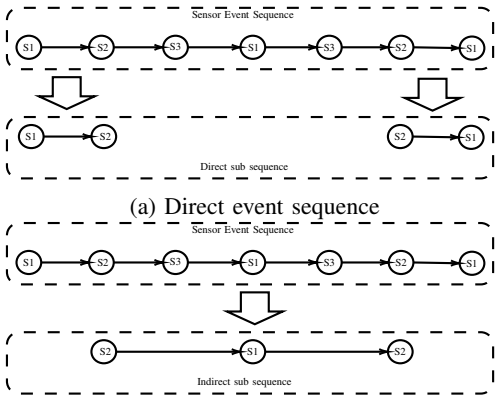
To estimate the location of sensors, the method first constructs a *sensor graph* $G_{sensor} = (S, E_{sensor}, T_{sensor})$, where S is a set of sensors deployed in a house, an edge $e_{sensor} = (s_i, s_j) \in E_{sensor}$ represents the fact that there was a direct trip between sensor s_i and sensor s_j, and $t_{s_i,s_j} \in T_{sensor}$ 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 [9], 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. In order to estimate E_{sensor}, 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). 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). For example, given the event sequence $seq_{event} = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), (s_1, t_4), (s_3, t_5), (s_2, t_6), (s_1, t_7)\}$. The *direct event sequence* of sensor pair (s_1, s_2) is $\{((s_1, t_1), (s_2, t_2)), ((s_2, t_6), (s_1, t_7))\}$ as in Fig 1a, and the *indirect event sequence* of sensor pair (s_1, s_2) is $\{((s_2, t_2), (s_1, t_4)), ((s_1, t_4), (s_2, t_6))\}$ as in Fig 1b.

To estimate the existence of an edge $e_{sensor} = (s_i, s_j) \in E_{sensor}$ between sensor s_i and sensor s_j, we define $size(direct(s_i, s_j))$ as the the number of *direct event sequence* of pair (s_i, s_j), and $size(indirect(s_i, s_j))$ as the number of *indirect event sequence* of (s_i, s_j). We calculate a likelihood $H(s_i, s_j)$ between a pair sensor (s_i, s_j) as equation 2. Then we assess the $H(s_i, s_j)$ by a threshold to generate the edge $e_{sensor} = (s_i, s_j)$.

$$H(s_i, s_j) = \frac{size(direct(s_i, s_j)}{size(direct(s_i, s_j) + size(indirect(s_i, s_j)} \quad (2)$$

Finally, the trip time $t_{s_i,s_j} \in T_{sensor}$ that the resident spends on walking on every edge $e_{sensor} \in E_{sensor}$ is 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 $X_{s_1,s_2} = \{t_2 - t_1, t_4 - t_2, t_6 - t_4, t_7 - t_6\}$ of pair (s_1, s_2). After generating X_{s_i,s_j}, the method has two steps.



(a) Direct event sequence

(b) Indirect event sequence

Fig. 1: Types of Event Sequence

a) The clustering step models different trip patterns (fast moving, typical trip, stop-and-go travel between location) from X_{s_1,s_2} in order to filter out irrelevant times.
b) The estimation step computes the trip time from the remaining clusters

a) The clustering: is based on the Gaussian Mixture Model (GMM), which is a probabilistic model that assumes all the trip times come from a mixture of a finite K Gaussian distributions. As exposed in Section III-A, the velocity of resident being composed of different behaviours, GMM is particularly well adapted to capture this complex distribution. Learning such a model from the data requires to determine the parameters π_k, μ_k and Σ_k of each component of the mixture. The probability of an observation X, is the linear combination of K components and can be represented as

$$p(X) = \sum_{k=1}^{K} \pi_k G(X|\mu_k, \Sigma_k) \quad (3)$$

where π_k is the mixing coefficient for the k-th component distribution. These parameters are estimated using expectation-maximization (EM). Figure 2 shows the results of the GMM learning using 4 components. It is clear from this figure that the GMM enables to separate short trip times from different sets of longer trip times.

b) The estimation: of the trip time $t_{s_i,s_j} \in T_{sensor}$ is then the median of the trip time instances belonging to the two first clusters (shortest trip time values). The clusters with the longest trip time values are assumed to due to stop-and-go behaviours or long displacements across the home.

D. Matching

In our previous work [9], we identified the sensors placed in key locations by searching for similar patterns in sensors data.

[1]Recall that the floor plan analysis is out of the scope of this paper.

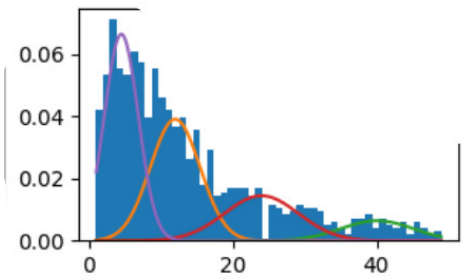


Fig. 2: Results of clustering of trip times between PIRs of the `table` and of the `living room` areas. Blue bars are the histogram of trip times while coloured Gaussian curves represent the weighted component distributions. X axis is in second and Y axis is the normalized ratio of the histogram.

Indeed, sensors situated very close to each other have a high probability to fire in the same way. When found, these similar patterns enabled to associate to the key sensor location, the closets sensors in the neighbouring area of the key location. It works well if every room deploys up to 2 sensors. However, this strategy is not adapted to scattered network when sensors do not share the same perception area.

In this study, unlike that method, every matching pattern is examined. 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_{l_u,l_v} from location l_u to location l_v by using the velocity in Table I, which is based on the recent studies about gait [13].

TABLE I: Velocity of Resident to Estimate Trip Time

Situation	Velocity (m/s)
Same Floor on open space	1.5
Separate rooms on the same Floor	1.3
Change floor	1

We denote $\mathcal{A} = \{A_1, A_2, ...\}$ is a set of all matching pattern where A_1, A_2 are the matching function $A : S \rightarrow L_{floor}$. By performing linear programming, we, also, introduce a feasible matrix $C_{feasible}(c_{iu}) \in \{0, 1\}$ being a matching possibility of sensor s_i can be located in location l_u. In specific, the sensors can be located on every location exept the sensor in key location. Then, the matrix C can be computed as in equation 4 by defining a set of sensor in key location $S_k \subset S$.

$$c_{ij} = \begin{cases} 0 & \textit{if } s_i \in S_k \wedge A : s_i \rightarrow l_u \text{does not exist} \\ 1 & \textit{otherwise} \end{cases} \quad (4)$$

We, also, introduce the matching matrix $M(m_{ij})$ in which m_{iu} is set to be 1 if we set the sensor s_i to match to location l_u ($A : s_i \rightarrow l_u$ is exist) otherwise is set to be 0. Then, we can assuss the matching matrix is validate by 2 constraints.

1) one sensor is mapped to one location.

$$\forall i \sum_{u=1}^{n} m_{iu} = 1$$

2) Each sensor is mapped to its feasible location.

$$\sum_{i=1}^{d} \sum_{u=1}^{n} m_{iu} c_{iu} = d$$

For each possible matching $A \in \mathcal{A}$, we calculate the matching score $score(A)$ where $A : s_i \rightarrow l_u$ and $A : s_j \rightarrow l_v$ by using equation 5.

$$score(A) = \sum_{\forall s_i, s_j \in S} \sum_{\forall l_u, l_v \in L_{floor}} m_{iu} m_{jv} (|t_{s_i,s_j} - t_{l_u,l_v}|)_{i \neq j} \quad (5)$$

We rank the matching score $score(A) \forall A \in \mathcal{A}$ and select the matching patterns whose the matching score $score(A)$ is lower than a slack value Δ_1. In specific, the slack value Δ_1 is calculated by the lowest matching score plus 10%. Then we generate the matching frequency matrix M_freqmf_{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 $l_u \in L_{floor}$ when the mf_{iu} is highest for each mf_{i*}.

IV. Experiment

A. Dataset

To evaluate the approach on set of scattered Infra-red sensors, we used the ContextAct@A4H dataset [14]. It is a rich, real-life daily living dataset collected in the Amiqual4Home[2] smart-home. This 87 m^2 Smart Home is equipped with home automation systems, multimedia controller, environmental meters and actuators, and means for observing human activity. As shown Figure 3, the kitchen and the living room are on the ground floor, the bedroom and the bathroom on the upper floor. 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.) connected through different protocols such as a KNX[3], UPnP (Universal Plug and Play) or X2D. The management of the home automation network, sending commands to the different actuators and receiving changes of sensor values, is operated through openHAB[4]. 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

[2]https://amiqual4home.inria.fr
[3]https://www.knx.org/
[4]https://www.openhab.org/

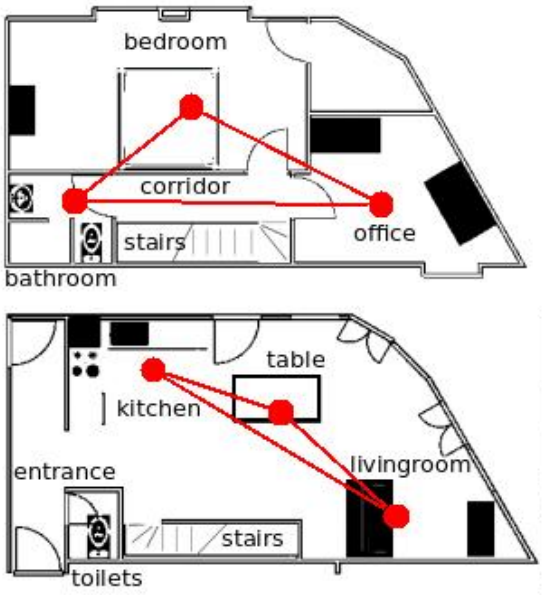


Fig. 3: The Amiqual4Home smart home plans with the PIR sensor network for each floor drawn in red. Top is upper flow, bottom is ground floor.

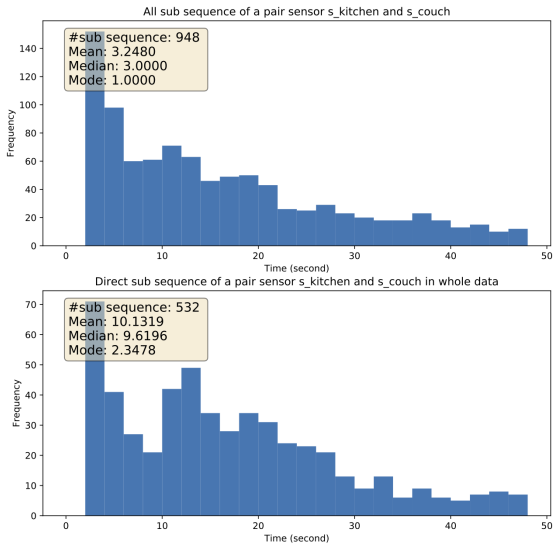


Fig. 4: Histogram of Trip time between kitchen and couch before clustering. Top is histogram of both direct and indirect event sequence, bottom is histogram of direct event sequence

(summer and fall respectively). This collection resulted in 30756 PIR firing.

B. Traveling Time

In this section, we show an analysis of the travelling time between each pair of sensors. In particular, we consider both *direct event sequence* and *indirect event sequence* to estimate which pair should be considered. According to Fig 4, the likelihood $H(s_i = kitchen, s_j = couch)$ is high, hence we assume the edge $e_{sensor} = (kitchen, couch)$ is exist.

For generating a trip time for edge e_{sensor}, we consider the trip time from both *direct event sequence* and *indirect event sequence*. According to Fig 3, kitchen and couch are close with each other. However, the table is located in the middle and resident can pass the sensor over the table when that resident walks from kitchen to couch. As in Fig 4, the trip time from *direct event sequence* will be incorrect (Median = 9.6). This is because the sensor at the living room's table will detect the movement of the resident when the resident walks from the kitchen to the living room's couch. As a result, the direct event sequence may not contain some moving events between kitchen and couch, and the estimated trip time become incorrect.

Table II shows the actual distances between a pair of rooms and estimated trip time by our method. The walking speed shows large variations but, apart from the travel `bathroom to office`, they stay within the range of human walking speed at home [13]. It can be noticed that the highest speed is the case within the same floor and the lowest is the case of floor transition. The stairs may play a role in this difference. Also, low speed can be observed from places where the resident may be seated (couch and table) or lying (bedroom).

TABLE II: Estimated Trip Time and Actual Distance between a Pair of Rooms

Pair		Estimated Trip Time(s)	Distance from G_{floor} (m)	Walking Speed (m/s)
bathroom	bedroom	6.00	7.03	1.17
bathroom	kitchen	8.00	10.43	1.30
bathroom	couch	16.50	18.16	1.10
bathroom	office	3.00	6.73	2.24
bedroom	kitchen	15.00	11.02	0.73
bedroom	office	7.00	4.98	0.71
kitchen	couch	3.00	5.45	1.82
kitchen	table	3.00	3.40	1.13
kitchen	office	14.00	10.72	0.77
couch	table	3.00	2.40	0.80
couch	office	16.00	18.45	1.15
average$^{\pm \text{standard deviation}}$				$1.18^{\pm 0.48}$

C. Matching performance

In this section, we analyze the matching between sensors and room locations. Table III shows the results of the association of each detected pattern. The raw accuracy 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.

TABLE III: Matching result

Actual	Estimated location										
	Bathroom	Bedroom	Kitchen	Living	Table	Office	Entrance	Toilet	Front Stairs	Stair	Corridor
Bathroom	4					2					
Bedroom		6									
Kitchen			6								
Couch					6						
Table				2			2	1	1		
Office	4					4					

V. 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 method is built on a previous work [9] which has been developed for dense network of sensors. The method presented in this paper has been adapted to deal with a more challenging and realistic situation of scattered and small set of sensors in a home. 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. Moreover, the smaller set of sensors increases the number of undecidable symmetries in the sensor graph. Given these difficulties, further work should concentrate on a better estimation of distance by modeling more precisely the velocity of the user according to her profile[13], her activities and the home topology. Our initial approach using Gaussian Mixture Models already showed to be sensible. Furthermore, the lack of available PIR sensors can be compensated with the inclusion of other sensors in the study such as switches or contact door detectors which are also related to displacement and activities. This particular aspect is the target of future work.

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.

References

[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] 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*, 2015, pp. 1101–1105.

[3] 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*, 2015, pp. 4:1–4:8.

[4] P. Chahuara, A. Fleury, F. Portet, and M. Vacher, "On-line Human Activity Recognition from Audio and Home Automation Sensors: comparison of sequential and non-sequential models in realistic Smart Homes," *JAISE - Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 4, pp. 399–422, 2016.

[5] 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.

[6] T. Miyazaki and Y. Kasama, "Multiple human tracking using binary infrared sensors," *Sensors*, vol. 15, no. 6, pp. 13 459–13 476, 2015.

[7] 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.

[8] N. Kitbutrawat, H. Yamaguchi, and T. Higashino, "Localization of binary motion sensors in house," in *13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017, Valencia, Spain, June 26-30, 2017*, 2017, pp. 1132–1137.

[9] N. Kitbutrawat, H. Yamaguchi, and T. Higashino, "Easytrack: Zero-calibration smart-home tracking system," *Journal of Information Processing*, vol. 27, pp. 1–11, July 2019.

[10] 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.

[11] A. Makki, A. Siddig, M. Saad, J. R. Cavallaro, and C. J. Bleakley, "Indoor localization using 802.11 time differences of arrival," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 614–623, March 2016.

[12] N. Twomey, T. Diethe, I. Craddock, and P. Flach, "Unsupervised learning of sensor topologies for improving activity recognition in smart environments," *Neurocomputing*, vol. 234, pp. 93 – 106, 2017.

[13] 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.

[14] P. Lago, F. Lang, C. Roncancio, C. Jiménez-Guarín, 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.