Occupancy Detection for Building Emergency Management Using BLE Beacons

Avgoustinos Filippoupolitis(✉), William Oliff, and George Loukas

Department of Computing and Information Systems,
University of Greenwich, London, UK
{a.filippoupolitis,w.oliff,g.loukas}@gre.ac.uk

Abstract. Being able to reliable estimate the occupancy of areas inside a building can prove beneficial for managing an emergency situation, as it allows for more efficient allocation of resources such as emergency personnel. In indoor environments, however, occupancy detection can be a very challenging task. A solution to this can be provided by the use of Bluetooth Low Energy (BLE) beacons installed in the building. In this work we evaluate the performance of a BLE based occupancy detection system geared towards emergency situations that take place inside buildings. The system is composed of BLE beacons installed inside the building, a mobile application installed on occupants' mobile phones and a remote control server. Our approach does not require any processing to take place on the occupants' mobile phones, since the occupancy detection is based on a classifier installed on the remote server. Our real-world experiments indicated that the system can provide high classification accuracy for different numbers of installed beacons and occupant movement patterns.

1 Introduction

Thanks to its exceptionally low power requirements, low cost and compatibility with most mobile devices and computers, Bluetooth low energy (BLE) is rapidly proving to be a very practical technology in e-health, sports, fitness, marketing in malls and other applications. We argue that its ability to provide proximity information with sufficient accuracy can extend its use in emergency management too, especially in buildings and other confined spaces, where traditional localisation technologies often fail. For example, having a mechanism to estimate the occupancy of different areas within a building can help emergency personnel produce a more optimal plan of action. In the literature on emergency management supporting technologies, it is often assumed that the emergency personnel or unmanned technical systems involved are aware of the locations where there are individuals requiring assistance/rescue [5,6,12], but this assumption can be highly inaccurate in many real-life situations. For example, during the 2015 terrorist attack in a Tunis museum, two tourists spent the night hiding in the museum only to be found the next day. Afraid to attract the attention of the terrorists, they had refrained from using their phones. BLE can

T. Czachórski et al. (Eds.): ISCIS 2016, CCIS 659, pp. 233–240, 2016.
DOI: 10.1007/978-3-319-47217-1_25

help both occupancy detection and indoor localisation, as has been acknowledged in a US Federal Communications Commission roadmap for BLE use in conjunction with WiFi to help locate 911 callers inside buildings.

There is a wide range of BLE based applications targeted to building occupants, including indoor navigation [7], activity recognition [1] and remote healthcare monitoring [11]. With respect to indoor occupancy estimation and localisation, we can find various approaches targeting different area types. The authors in [4] discuss the use of Apple's iBeacon protocol for building occupancy detection. They evaluated their approach using a single room and predicted whether the occupant was inside or outside. A system that detects the locations of occupants inside an office is presented in [2]. This is used to control a building management system that the authors evaluate inside an office area. The estimation of a building's occupancy using Arduino based beacons is described in [3]. The authors evaluate the system by estimating an occupant's presence inside or outside a single room. The authors in [9] employ iBeacons inside the floor of a building in order to evaluate the performance of an occupancy estimation system for hospitals. Their system has a high overall accuracy but there are no accuracy results for individual areas. In [10] the authors propose an indoor localisation system that uses BLE beacons inside an office building. Their approach achieves a high localisation accuracy (for 75 % of the time the localisation error is lower than 1.8 m) however they have not evaluated the effect of walking speed or beacon locations. Finally, the authors in [8] propose an indoor localisation system based on BLE beacons. The system is evaluated inside a single room and although they claim a high accuracy rate, their results are limited.

Our approach is targeted towards emergency situations and aims to provide an estimate of the number of occupants inside areas such as offices, laboratories and conference rooms. Even if our proposed system stops functioning (e.g., due to a natural or man-made disaster), it is still able to provide very useful information related to the spatial distribution of the occupants at the time before the incident took place.

2 Description of the System

Our approach is based on the use of BLE beacons located inside the building that communicate with a mobile application installed on the occupant's phone. The beacons use a non-connectible mode, the BLE advertising mode, to periodically broadcast advertisement packets that include information such as the beacon's unique ID. A mobile phone located in the vicinity of a beacon receives the packets and processes them using a mobile application. In a commercial setting the main assumption is that the mobile application has knowledge of the beacons' location inside the building and of the mapping between beacons and rooms or areas. This information is then used by the mobile application, in conjunction with the received BLE packets, in order to calculate the user's location inside the building. Finally, the mobile application sends its location to a remote server which then replies with contextual information (such as a targeted micro-location based advertisement).

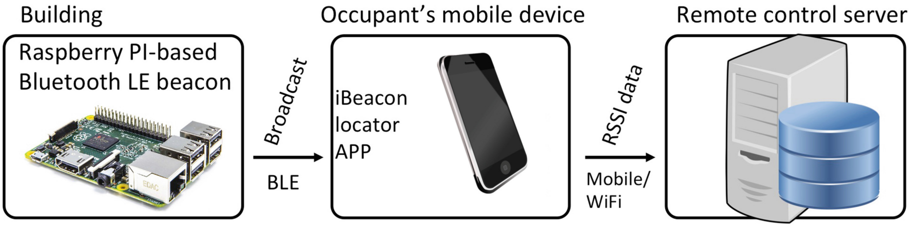


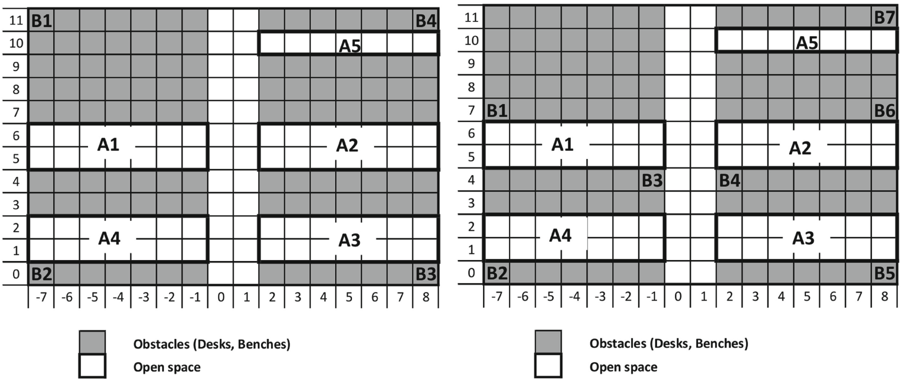
Fig. 1. System architecture

Figure 1 illustrates our system's operation in an emergency situation that takes place inside a building. The mobile application installed on the occupants' phones receives BLE messages from multiple beacons. It then sends their RSSI values and respective beacon IDs to the remote control server. Finally the server, upon reception of this information from a mobile device, uses a trained classifier to update the building occupancy estimation. Our approach has numerous advantages. Firstly, the mobile phone does not need to know the mapping between beacon ID and location of beacons inside the building. Also, the mobile phone does not process the received beacon packets to calculate its location and the remote control server does not send information back to the mobile phone. Since our system does not involve localisation related processing by the mobile application, we can use mobile devices that have low computational power and memory capacity. The remote control server is responsible for processing the data that the mobile application sends and for calculating the building occupancy. To achieve this, we conduct a single data gathering phase during which the data gathered are used to train a classifier. Section 3 provides further details on this process. After the data gathering phase has been completed, the system is able to operate in normal mode as shown in Fig. 1.

For our BLE beacons we used a Raspberry Pi 2 with a Bluetooth 4 BLE USB module. We implemented the iBeacon protocol, which is the BLE beacon implementation proposed by Apple. By using an open platform such as the Raspberry Pi, we avoided the limitation of being tied to a specific beacon manufacturer. To identify the iBeacons, we used a Universally Unique Identifier (UUID), a major number and a minor number for each of them. The UUID is used to separate the iBeacons being used in our experiments from other unassociated Bluetooth devices. The major number is used to define local groups of iBeacons (e.g. belonging to a certain building or floor) and the minor number is used to define each individual iBeacon within a local group. We can use our Android mobile application for the data gathering phase as well as for the normal operation of the system. When the mobile application receives a BLE advertising data packet from an iBeacon during the data gathering phase, it extracts and logs the UUID, major number, minor number and transmission (Tx) power of the beacon from the packet's payload. The application also logs the received signal strength indicator (RSSI) of each received BLE packet. Finally, an area label is manually assigned to each packet by the user based on his actual location inside the building. Under normal operation mode, the mobile application simply receives BLE packets from beacons and sends their RSSI values and respective beacon IDs to the server. The remote control server processes the data sent by the mobile application in order to calculate the occupancy of the building. In normal operation mode, the server receives information from the mobile application running on an occupant's mobile phone and uses a trained classifier to update the building occupancy estimation. The training of this classifier is performed during the initial data gathering phase. We must note, however, that it is not necessary for the training to take place in the server. The only requirement is that the trained classifier model is stored on the server so that it can be used during normal operation.

3 Experimental Evaluation

We evaluated the performance of our system in the computer laboratory of the University of Greenwich. This is essentially an office space that includes objects such as desks, benches, computers, panels and chairs. We have identified five areas inside the laboratory (A1-A5), as illustrated in Fig. 2. An orthogonal grid was used to map the experimental area, with each grid square equal to an area of 1 m^2. We investigated two beacon deployment configurations: one involving four beacons and one involving seven beacons, as shown in Figs. 2(a)–(b). For the data gathering phase, we used our mobile application in data gathering mode. The beacons' transmission frequency was set to 8 Hz and their transmission power to 4 dBm. To increase the level of realism, instead of standing inside each area we moved according to a "Walk and Stop" pattern that involved spending 10 s on each grid point before moving to the next one. For each BLE packet received the mobile application logged the UUID, major number, minor number and RSSI



(a) 4 beacons (B1-B4)

(b) 7 beacons (B1-B7)

Fig. 2. Experimental area and beacon positions for the two different configurations

and we assigned an area label (A1 to A5) based on our actual location. For each of the two beacon setups, we conducted two runs of this data gathering phase. This resulted in a dataset size of over 44,000 packets for the 4 beacon setup and of over 78,000 packets for the 7 beacon setup.

We modelled our problem as a multi-class classification problem, with the number of classes equal to the number of areas in our environment (i.e. five classes). Our raw dataset contained individual packets coming from specific beacon IDs, with a respective RSSI value and an area label. To transform this to a dataset that can be used to train a classifier, we used a data segmentation approach involving a non-overlapping sliding window. For each beacon inside a specific area, we calculated the average and the standard deviation of its RSSI over the window samples and used these as the features of our classification problem. For the four beacon setup, this resulted in eight features while for the seven beacon setup we had fourteen features. For our classifier we have chosen a support vector machine with radial basis function kernel (SVM). The reason behind this choice is that SVMs can successfully deal with non-linearly separable data. We partitioned the dataset into 80 % training set and 20 % test set and used 10-fold cross validation for hyper-parameter tuning. We used a confusion matrix for presenting our classification results, where its rows represent the instances in an actual class and its columns the instances in a predicted class. The values of the matrices are normalised by the number of elements in each class, to better illustrate the classification accuracy for each class.

3.1 Results for “Walk and Stop” Scenario

Figure 3 illustrates our classification results for the “Walk and Stop” scenario and the four beacons setup. In the case of a 0.5 s window, we can observe that the classification accuracy ranges from 64 % to 89 %. Increasing the window size to 1 s, as depicted in Fig. 3(b), improves the classification performance especially for Area 2 where its classification accuracy has now increased from 64 % to 81 %. Further increasing the window size to 2 s, as shown in Fig. 3(c), does not provide a clear improvement of the classification accuracy. For example, although Area 1 is now classified with 100 % accuracy, the performance of the classifier for Area 2 has dropped to 68 %. By inspecting Figs. 3(a)–(c) we can observe a consistently low performance of our classifier with respect to Area 2. This can be explained if we look at the spatial distribution of beacons with respect to areas, as depicted in Fig. 2(a). We can observe that the number of beacons is less than the number of areas (four versus five respectively). Moreover, each Area can be associated with one specific beacon which is closest to it: Area 1 with Beacon 1, Area 4 with Beacon 2, Area 5 with Beacon 4 and Area 3 with Beacon 3. However, there is no one Beacon that can be associated with Area 2. The two closest beacons to Area 2 are Beacon 4 and Beacon 3. This sparse beacon deployment explains the low classification performance for Area 2. We can also verify from Figs. 3(a)–(c) that Area 2 is consistently misclassified as Area 3 or Area 5, which are the two areas closest to Beacon 3 and Beacon 4.

	A1	A2	A3	A4	A5
A1	0.82	0	0.04	0.13	0.01
A2	0	0.64	0.17	0.01	0.17
A3	0.03	0.11	0.8	0.05	0
A4	0.16	0.01	0.01	0.81	0
A5	0.04	0.07	0	0	0.89

(a) Window=0.5 s

	A1	A2	A3	A4	A5
A1	0.86	0.02	0	0.04	0.09
A2	0	0.81	0.12	0	0.06
A3	0.02	0.22	0.76	0	0
A4	0.12	0.03	0	0.82	0.03
A5	0.03	0.11	0	0	0.86

(b) Window=1 s

	A1	A2	A3	A4	A5
A1	1	0	0	0	0
A2	0	0.68	0.16	0	0.16
A3	0.04	0.24	0.72	0	0
A4	0.09	0	0	0.91	0
A5	0.05	0.05	0	0	0.9

(c) Window=2 s

Fig. 3. Confusion matrices for SVM, using 4 beacons and different window sizes ("Walk and Stop" Scenario)

	A1	A2	A3	A4	A5
A1	0.92	0	0.01	0.06	0
A2	0.01	0.92	0.01	0	0.05
A3	0.02	0.06	0.88	0.02	0.01
A4	0.05	0	0	0.94	0.01
A5	0	0.04	0.1	0	0.86

(a) Window=0.5 s

	A1	A2	A3	A4	A5
A1	1	0	0	0	0
A2	0	0.95	0.05	0	0
A3	0.05	0.02	0.91	0	0.02
A4	0.06	0	0.03	0.91	0
A5	0	0.02	0.05	0	0.93

(b) Window=1 s

	A1	A2	A3	A4	A5
A1	0.94	0	0	0.06	0
A2	0	1	0	0	0
A3	0.1	0	0.9	0	0
A4	0.04	0	0	0.96	0
A5	0	0	0	0	1

(c) Window=2 s

Fig. 4. Confusion matrices for SVM, using 7 beacons and different window sizes ("Walk and Stop" Scenario)

By increasing the number of beacons to seven, we observed a significant improvement in the classification accuracy for all window sizes, as depicted in Fig. 4. For a window size of 0.5 s the classification accuracy ranges from 86 % to 94 %, as shown in Fig. 4(a). Figure 4(b) illustrates the results for a window size equal to 1 s. We can verify that increasing the window size improves the classification accuracy, which now ranges from 91 % to 100 %. Finally, further increasing the window size to 2 s does not yield a significant improvement in accuracy, as Fig. 4(c) shows. We should also note that in the seven beacon configuration we do not observe the consistent misclassification of Area 2, as was the case in the four beacon configuration.

3.2 Results for "Random Walk" Scenario

To investigate the effect of the movement pattern on the classification accuracy, we have conducted an additional experiment with the seven beacon configuration. This time, we moved inside each area without stopping on grid points. The movement involved randomly choosing a destination grid square point within each area, walking towards it, then choosing another one and repeating the same procedure for each area. The total duration of this "Random Walk" scenario was equal to that of the "Stop and Walk" scenario for the seven beacon configuration, in order to achieve the same dataset size.

As we can observe from Fig. 5, the classification accuracy is lower compared to the one shown in Fig. 4. For a window size of 0.5 s, the accuracy ranges

	A1	A2	A3	A4	A5
A1	0.96	0.01	0	0.02	0
A2	0.02	0.88	0.05	0	0.05
A3	0.01	0.08	0.84	0.02	0.04
A4	0.02	0.02	0.04	0.9	0.01
A5	0	0.08	0.03	0.01	0.87

(a) Window=0.5 s

	A1	A2	A3	A4	A5
A1	0.98	0	0	0.02	0
A2	0.03	0.97	0	0	0
A3	0	0.05	0.85	0.05	0.05
A4	0.1	0	0.05	0.85	0
A5	0	0.05	0	0	0.95

(b) Window=1 s

	A1	A2	A3	A4	A5
A1	0.95	0	0	0.05	0
A2	0	0.94	0.06	0	0
A3	0	0.04	0.81	0.08	0.08
A4	0	0	0	1	0
A5	0	0	0	0.06	0.94

(c) Window=2 s

Fig. 5. Confusion matrices for SVM, using 7 beacons and different window sizes ("Random Walk" Scenario)

between 84 % and 96 %. Increasing the window size from 0.5 s to 1 s results in an improvement in accuracy which ranges between 85 % and 97 %. A window size of 2 s improves the classification accuracy further, especially for Area 4 which increases to 100 % from the 85 % of the 1 s window case.

This was expected, as the constant movement of the occupant in the "Random Walk" makes training the system more challenging, resulting in reduced accuracy compared to the more static "Walk and Stop" case. At the same time, increasing the size of the window results in averaging RSSI values over a longer time interval for each data point. This compensates for the constant movement of the occupant but reduces the responsiveness of the system, because under normal system operation the server would have to wait for 2 s before receiving RSSI data from the mobile application.

4 Conclusions and Future Work

In this work, we have evaluated the performance of a BLE based occupancy detection system geared towards emergency situations that take place inside buildings. The system is composed of BLE beacons installed inside the building, a mobile application installed on occupants' mobile phones and a remote control server located outside the building. We do not require any localisation calculations to take place on the mobile phone, since the occupancy detection is based on a classifier installed on the remote server. Our real-world experiments indicated that the system can provide a high classification accuracy for different beacon deployment configurations and movement patterns of the building occupants. In future work, we will investigate a greater range of occupant walking speeds and beacon deployment configurations. We also plan to study how our system's performance is affected by different beacon transmission frequencies. Finally, we believe it is worth investigating the use of machine learning algorithms based on neural networks and deep learning to evaluate whether they can further improve the classification accuracy of our system.

References

1. Alam, M., Pathak, N., Roy, N.: Mobeacon: an iBeacon-assisted smartphone-based real time activity recognition framework. In: Proceedings of the 12th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (2015)
2. Choi, M., Park, W.K., Lee, I.: Smart office energy management system using bluetooth low energy based beacons and a mobile app. In: 2015 IEEE International Conference on Consumer Electronics (ICCE), pp. 501–502. IEEE (2015)
3. Conte, G., De Marchi, M., Nacci, A.A., Rana, V., Sciuto, D.: BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system. In: BuildSys@ SenSys, pp. 11–19 (2014)
4. Corna, A., Fontana, L., Nacci, A., Sciuto, D.: Occupancy detection via iBeacon on Android devices for smart building management. In: Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, pp. 629–632. EDA Consortium (2015)
5. Dimakis, N., Filippoupolitis, A., Gelenbe, E.: Distributed building evacuation simulator for smart emergency management. Comput. J. **53**(9), 1384–1400 (2010)
6. Filippoupolitis, A., Gorbil, G., Gelenbe, E.: Spatial computers for emergency support. Comput. J. **56**(12), 1399–1416 (2012)
7. Fujihara, A., Yanagizawa, T.: Proposing an extended iBeacon system for indoor route guidance. In: 2015 International Conference on Intelligent Networking and Collaborative Systems (INCOS), pp. 31–37. IEEE (2015)
8. Kajioka, S., Mori, T., Uchiya, T., Takumi, I., Matsuo, H.: Experiment of indoor position presumption based on RSSI of Bluetooth LE beacon. In: 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE), pp. 337–339. IEEE (2014)
9. Lin, X.Y., Ho, T.W., Fang, C.C., Yen, Z.S., Yang, B.J., Lai, F.: A mobile indoor positioning system based on iBeacon technology. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 4970–4973. IEEE (2015)
10. Palumbo, F., Barsocchi, P., Chessa, S., Augusto, J.C.: A stigmergic approach to indoor localization using bluetooth low energy beacons. In: 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE (2015)
11. Sugino, K., Katayama, S., Niwa, Y., Shiramatsu, S., Ozono, T., Shintani, T.: A bluetooth-based device-free motion detector for a remote elder care support system. In: 2015 IIAI 4th International Congress on Advanced Applied Informatics (IIAI-AAI), pp. 91–96. IEEE (2015)
12. Timotheou, S., Loukas, G.: Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios. In: Proceedings of the 2009 ACM Symposium on Applied Computing, pp. 1171–1175. ACM (2009)