



IBeaconMap: Automated Indoor Space Representation for Beacon-Based Wayfinding

Seyed Ali Cheraghi^(✉), Vinod Namboodiri, and Kaushik Sinha

Department of Electrical Engineering and Computer Science,
Wichita State University, Wichita, KS, USA
sxcheraghi@shockers.wichita.edu,
{vinod.namboodiri,kaushik.sinha}@wichita.edu

Abstract. Traditionally, there have been few options for navigational aids for the blind and visually impaired (BVI) in large indoor spaces. Some recent indoor navigation systems allow users equipped with smartphones to interact with low cost Bluetooth-based beacons deployed strategically within the indoor space of interest to navigate their surroundings. A major challenge in deploying such beacon-based navigation systems is the need to employ a time and labor-expensive beacon planning process to identify potential beacon placement locations and arrive at a topological structure representing the indoor space. This work presents a technique called IBeaconMap for creating such topological structures to use with beacon-based navigation that only needs the floor plans of the indoor spaces of interest.

Keywords: Indoor wayfinding · Accessibility · Machine learning · Internet of things (IoT) · Computer vision

1 Introduction and Related Works

Recent advances in global positioning systems (GPS) and mapping technologies provide accurate and simple to use means for wayfinding for outdoor environments. For indoor environments, reading and following signs remains the easiest and most reliable option because GPS and associated advances for outdoor environments do not apply. This has, however, meant that indoor wayfinding has remained a challenge for the blind and visually impaired (BVI) in our society. Indoor environments can be geographically large and intimidating such as grocery stores, airports, sports stadiums, large office buildings, and hotels. A solution to the indoor wayfinding problem for the BVI also has broad applications for the sighted population. Recent work has developed a system of wayfinding for the BVI using low-cost, stamp-size Bluetooth Low Energy (BLE) “beacon” devices embedded in the environment [1, 3, 6] that interact with smartphones carried by users. Such beacon-based navigation systems have achieved promising preliminary results indicating that they may be a viable solution for indoor

wayfinding for the BVI if some of the underlying challenges to the deployment of such systems can be solved.

Exploring an indoor space and its information for someone who is blind requires knowing the information somehow beforehand and have it presented when the person approaches the proximity of a Point of Interest (POI). This information can be entered manually by a human about the entire indoor space which is the current and traditional approach to beacon planning. Manual determination of all beacon placement locations and path computations is time-consuming and labor-expensive, especially for large indoor spaces. Such an approach requires the manual identification of walking paths on a floor plan, marking of points of interest, determining the distance between any two points of interest, determining the orientation between them for navigation, computing shortest paths between points of interests, and subsequent adjustments to optimize the resulting paths that may require further iterations of the entire process.

None of the efforts so far had designed a fast, and largely automated method for representing indoor spaces as topological structures for accurate and timely beacon-based navigation. Such a method (as proposed in this work) will benefit all current efforts in deploying indoor beacon-based navigation systems. Work related to creating representations of indoor spaces have been around for a while (e.g. [7, 8]). These can mainly be differentiated based on the approach used in collecting the required information and in the techniques employed to create the desired representations. Prior work such as in [2] can help create indoor space representation through floorplans which serve as inputs to IBeaconMap. IBeaconMap differs from this class of work by taking files in simple image formats or PDFs as input and employing a combination of computer vision and machine learning techniques. In addition to marking points of interests on floor plans as beacon locations, IBeaconMap can also mark strategic points such as intersections which are important for BVI navigation. None of the previous work on indoor space representations focused on providing outputs catering to the special needs for beacon-based wayfinding that include beacon location markings, indoor paths connecting these locations, a weighted connectivity graph as topological structure representation, and directional orientations for paths. The web-based mapping tool developed as part of NavCog [1], the only other tool with a similar objective as IBeaconMap, requires a user to mark all beacon locations and walking paths first on a floor plan image. This higher-level of manual involvement is expected to not scale well thus rendering the tool not as desirable in many situations.

Additional approaches for indoor space representations beyond extraction from architectural floor plans are that of robotic mapping and crowdsourced approaches [9]. Robotic mapping approaches are likely to be more expensive to implement and time-intensive while crowdsourced approaches, although inexpensive and maybe even free, will not be as accurate or fast as IBeaconMap. Further, the recent work on using crowdsourcing to deploy beacons in [5] assumes that beacon locations are already known; thus, IBeaconMap could be a useful first tool to create location markings where beacons can then be placed in a crowdsourced fashion.

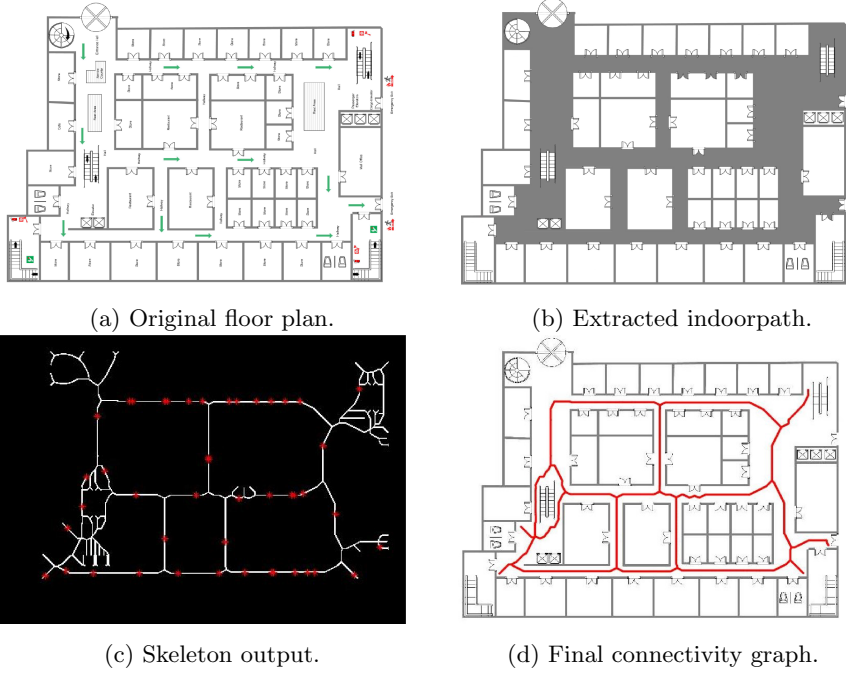


Fig. 1. Outputs from execution of IBeaconMap on a floor plan.

2 Overall Approach

A floor plan image's analysis is performed in four phases:

Phase 1: Indoor path identification - The goal of phase 1 is to extract the indoor path and adjoining POIs from the floor plan. The walking path connects all the building blocks (doors, stairs etc.) to each other, so finding it first makes it easier to find POIs. Furthermore, having the indoor path helps find the shortest path from any office or point of interest to any other. Walking paths were found by identifying the largest contiguous block of pixels within the indoor space; this contiguous block of pixels has to be the walking path with all other areas within the floor plan having disconnections due to doors, walls, stairs etc. The largest contiguous area¹ is then labeled so that it can be marked off as the walking path. Figure 1b demonstrates the original floor plan with the gray area added manually for illustration purposes to show the indoor path in the original image.

Phase 2: Building block detection - After obtaining the foreground, the next step is to find all the required building blocks in a floor plan and get the specific coordinates of their locations. To achieve this goal, IBeaconMap provides three different approaches: feature detection and matching (FDM), feature

¹ Any area to be added or removed from consideration as a restricted area can be marked as such in floor plan pre-processing steps.

detection, matching, and Support Vector Machine (SVM) [4] as one approach to Supervised Machine Learning (FDM + SML), and feature detection and supervised machine learning (FD + SML). The reason to use three different techniques is to provide options to users when faced with varying quality and complexities of floor plans supplied as input. The FDM approach is the fastest of the three, and is very accurate if the provided floor plans are of high resolution and without a high density of features. If the provided floor plan does not meet this criteria, as is possible when using scanned images of floor plan drawings made many decades ago, the accuracy can suffer. Having the other two approaches besides FDM provides more opportunities to arrive at an acceptably accurate result. The addition of SML to FDM allows removing some false positives from the FDM approach output, helping improve accuracy. For cases where FDM is expected to have very high inaccuracies, it can be skipped altogether. Instead, a pre-processing step of FD can be executed to first collect all possible features in the floor plan (a computationally intensive step) followed by the SVM to classify building blocks with reasonable accuracy. After obtaining building blocks' locations (POIs) in terms of (x, y) pixels using one of the above techniques, the next step is to find the available path between them.

Phase 3: Skeleton generation - To connect one POI to another, a path is required that does not pass through a wall, stair or any point which has a color other than white (after the floor plan is converted to a binary image). Since the locations of identified building blocks can be on the black line or be blocked in some ways, we desire to map them onto specific pixels of the indoor path already found. To achieve this, the boundary pixels of the indoor path are removed without letting the indoor path break apart. Then by using euclidean distance, the closest points on the indoor path skeleton to the building blocks are located (Fig. 1c).²

Phase 4: Connectivity graph generation - After mapping building blocks³ on the indoor path skeleton, we need to find the paths connecting any POIs which will lead to creating a connectivity graph on which path computations for navigation can be performed. To determine one-hop path distances between POI's, the IBeaconMap algorithm considers the indoor path skeleton to be the only non-zero pixels in the floor plan image. This by itself does not provide the one-hop paths between POIs, but the skeleton can be traversed in a breadth-first fashion beginning from a POI pixel by pixel to find various features. The connectivity graph arrived at for the example floor plan under consideration is shown in Fig. 1d.

² The function `bwmorph` used comes from the equivalent MATLAB function that was used to perform the skeleton generation operation.

³ If there were errors in building block detection, post-processing steps can correct them; after such steps the connectivity graph generation phase is executed again.

3 Evaluation

Two metrics were chosen to show the effectiveness of IBeaconMap. The first metric is that of accuracy of IBeaconMap’s output in terms of the number of beacon locations correctly identified versus those that were incorrect. The incorrect ones are further broken down into beacon locations that were missed and those that were redundantly added. A correct identification of a beacon location involves finding a POI and intersections. A visual comparison of beacon location marking outputs from a manual beacon planning process is also presented to provide a visual sense of accuracy of IBeaconMap. A manual approach is expected to be the most accurate as a skilled human can best determine where a beacon should be placed through an on-site survey. The question to us was “is there a way for us to significantly automate the beacon planning process while preserving as much of the accuracy of the manual process?”. The second metric is the processing time for IBeaconMap to take a floor plan as input and produce its output. This metric is thus a measure of the reduction in time and labor in arriving at beacon locations and connectivity graph for indoor navigation. Any manual post-processing required to fix inaccuracies would need to be added onto this time for a fairer comparison with a completely manual process; however, the aim with IBeaconMap was to keep the manual corrections to be minimal.

3.1 Basic Results

The basic results are from the FDM scheme, which is typically the recommended scheme (due to its low processing time) unless the floor plans have low resolution or very high density of building blocks. The basic results presented here use the indoor walking path detection of building blocks only option as this option is expected to be more commonly used.

Figure 2 shows the beacon location marking results using FDM on two floor plans that were of high resolution; one from a shopping mall and the other a small research building with offices and laboratories. For each floor plan, beacon locations identified by a manual process (finding all POIs visually and marked) are shown along with those generated by IBeaconMap. It can be seen that the outputs are remarkably accurate. The major difference can be seen as the small mismatch in locations at each point of interest such as doors, stairs etc. which accounts for over half of an deviations seen and is easily correctable. This mismatch was because the manual process intentionally marks beacon locations on the side of a door or stairs while IBeaconMap marks them at the center leaving those who deploy the beacons to make the decision as to which side to place the beacon. In addition IBeaconMap marks additional locations at intersections which would be very useful during navigation. If two POIs are very close to each other, within a distance of c meters (we used $c = 2$ m in this paper), IBeaconMap just affixes one beacon location that can serve both points. Some POIs that perhaps would have been omitted as beacon locations during a manual process (due to knowledge that those POIs will not be useful), are marked by IBeaconMap in the shown image; such location can be removed manually.

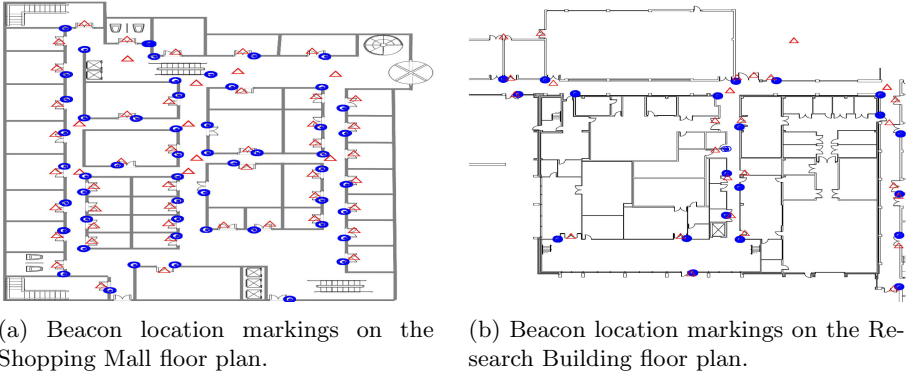


Fig. 2. Beacon locations as output on two different floor plans provided as input. Blue solid circles indicate beacon locations marked by a manual process while red triangles show beacon locations from IBeaconMap. (Color figure online)

The computation time for both indoor floor plans considered in Fig. 2 were analyzed on an Intel i5-5200U CPU (2.20 GHz) with 8 GB RAM on a 64-bit Windows 10 OS. The Shopping Mall floor plan took 15.64 s in total to provide the final outcome while Research Building only took 22.07 s. The computation for the former is faster as it has fewer building blocks/features that need to be detected next to the indoor path. Thus, these typical floor plans can be analysed to not only provide beacon locations to use, but they also provide a connectivity graph for navigation in under 1 min. An entire building with multiple floors thus can be analyzed (and generation of connectivity information and beacon locations) in an automated fashion in the order of minutes to a few hours depending on its size. A manual process, that involved drawing walking paths, marking beacon locations, measuring and entering graph data structure connectivity information, weights, and directional orientations as experienced by the authors for the research building floor plan in [3], took over 1 h to arrive at similar outcomes; larger buildings with many more POIs would have taken many more hours if not days per floor. It is important to remember that there may be post-deployment alterations required for which an automated tool again can make changes easier.

3.2 Comparison of Building Block Detection Techniques

Evaluation results from four different floor plans are shown here. Many other floor plans were analyzed and tested to ensure that the results shown here are representative of a larger trend. The first two (Research Building and Shopping Mall) are those already seen in Figs. 2a and 2b. An additional two, called Large Area and Scanned Image respectively, were added. The Large Area floor plan is of a 75,000 sq. ft indoor facility with a large number of potential POIs, some of which are densely congregated as well. The fourth floor plan was the same as the

first (Research Building), but a low-resolution (200 dpi) scanned image. These Large Area and Scanned Image floor plans were used to test the worst case for FDM and see how the SML based algorithms helped in such cases.

Table 1. Comparison of Building Block Detection Techniques - Indoor Path Only

		Correct	Incorrect		Processing Time (s)			Correct	Incorrect		Processing Time (s)
			Missed	Redundant					Missed	Redundant	
Research Building	FDM	30	1	3	22.07	Large Area	FDM	105	1	23	173.62
	FDM+SML	29	2	2	40.5		FDM+SML	98	8	16	282.83
	FD+SML	31	0	12	51.94		FD+SML	105	1	18	285.79
Shopping Mall	FDM	49	0	12	15.64	Low-Resolution Image	FDM	22	9	6	14.46
	FDM+SML	49	0	4	41.46		FDM+SML	21	10	3	26.1
	FD+SML	49	0	1	44.41		FD+SML	31	0	15	38.80

Table 2. Comparison of Building Block Detection Techniques - Full Floor Plan

		Correct	Incorrect		Processing Time			Correct	Incorrect		Processing Time
			Missed	Redundant					Missed	Redundant	
Research Building	FDM	63	4	16	11.30	Large Area	FDM	237	11	97	185.73
	FDM+SML	60	7	14	31.87		FDM+SML	233	15	59	351.38
	FD+SML	67	0	9	67.94		FD+SML	248	0	68	875.97
Shopping Mall	FDM	49	0	40	18.77	Low-Resolution Image	FDM	42	25	30	6.6
	FDM+SML	49	0	27	25.18		FDM+SML	42	25	20	21.3
	FD+SML	49	0	0	81.33		FD+SML	67	0	13	45.99

Indoor path detection only - The results for the detection along indoor walking path only is shown in Table 1. It can be seen that all three building block detection schemes perform with a high accuracy in terms of correctly identifying POIs with very few missed detections. The fast FDM scheme does very well for the smaller and simpler floor plans (Research Building and Small Area) and looks adequate for such cases. The FD + SML scheme helps improve detection accuracy significantly in the case of the low-resolution scanned image where FDM does not do well. The FD + SML scheme also seems to work better than FDM for floor plans with high density as in the Large Area floor plan. The FDM + SML scheme acts primarily as an “enhancer” to the FDM scheme, helping reduce some of the redundant locations identified, sometimes however at the cost of adding some more to missed detections. All schemes have some redundant identifications (false positives) which will need to be “scrubbed off” through a post-processing step as shown in Fig. 3. In terms of processing time, FDM was the fastest and FD + SML typically took the most time.

Full floor plan detection - The results for the entire floor plan building block detection is shown in Table 2. This being the worst case for building block detection due to the presence of multiple layers, it can be seen that the number of redundant beacon locations identified are larger; however, most POIs are still correctly identified. The FD + SML scheme again improves upon that of the FDM scheme when image resolution is poor or has a high density of POIs. The relative processing times of each scheme remains the same as in the indoor path only case, except that there is an overall increase due to the consideration of the entire floor area. As the floor plan area increases (as in the Larger Area floor plan), the FD + SML scheme processing time does increase faster than the other schemes due to its need to execute its three step process.

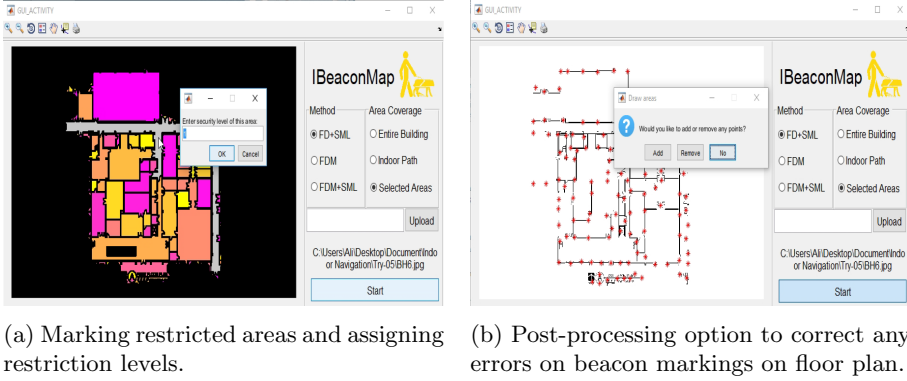


Fig. 3. Snapshots of IBeaconTool options.

4 Conclusion

This work presented a largely automated technique called IBeaconMap to prepare an indoor space for beacon-based wayfinding for the BVI and other sighted users. Such a technique solves the current challenge of creating indoor space representations in a time and labor-efficient manner. Evaluations show IBeaconMap to be fast computationally and reasonably accurate (depending on input resolution and space characteristics) thus presenting itself as a scalable tool in preparing all indoor spaces for beacon-based wayfinding in the future.

Acknowledgement. This work has been supported by NSF award #1951864.

References

1. Ahmetovic, D., Gleason, C., Ruan, C., Kitani, K., Takagi, H., Asakawa, C.: Navcog: A navigational cognitive assistant for the blind. In: International Conference on Human Computer Interaction with Mobile Devices and Services. ACM (2016)
2. Alzantot, M., Youssef, M.: Crowdsinside: Automatic construction of indoor floor-plans. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems, pp. 99–108 (2012)
3. Cheraghi, S.A., Namboodiri, V., Walker, L.: Guidebeacon: Beacon-based indoor wayfinding for the blind, visually impaired, and disoriented. In: IEEE Pervasive Communications (PerCom), pp. 121–130 (2017)
4. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
5. Gleason, C., Ahmetovic, D., Toxtli, C., Savage, S., Bigham, J.P., Asakawa, C.: Luzdeploy: A collective action system for installing navigation infrastructure for blind people. In: Web For All. ACM (2017)
6. Kim, J.E., Bessho, M., Kobayashi, S., Koshizuka, N., Sakamura, K.: Navigating visually impaired travelers in a large train station using smartphone and bluetooth low energy. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, SAC 2016. pp. 604–611. (2016)

7. Krūminaitė, M., Zlatanova, S.: Indoor space subdivision for indoor navigation. In: Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. pp. 25–31. ISA '14 (2014)
8. Niua, L., Song, Y.: A schema for extraction of indoor pedestrian navigation grid network from floor plans. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. vol. XLI-B4. Prague, Czech Republic (July 2016)
9. Zhang, Y., Liu, J., Hoffmann, G., Quilling, M., Payne, K., Bose, P., Zimdars, A.: Real-time indoor mapping for mobile robots with limited sensing. In: The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010). pp. 636–641 (Nov 2010)