Using Personnel Movements For Indoor Autonomous Environment Discovery

R. K. Harle A. Hopper
rkh23@eng.cam.ac.uk hopper@eng.cam.ac.uk
Laboratory for Communication Engineering
Department of Engineering
University of Cambridge, UK

Abstract

We present a novel method of extracting topological and metric geographical data using only positional data sensed from personnel movements. We extend research from the field of robotics to cope with the gross non-uniformity of sightings that is characteristic of real people in an indoor environment, and any unintentional obstruction of positioning by the user.

We use real data collected using the Bat positioning system installed in the Laboratory for Communication Engineering to present the results of implementing the method. We successfully derive useful information from the data, and suggest further ways in which the techniques described are useful in a ubiquitous, sensor-driven computing environment.

1. Introduction

In ubiquitous computing, we are beginning to see the emergence of high accuracy personnel positioning systems [9, 15, 16, 18, 20, 22, 23] which are vital enabling technologies for the context-aware computing paradigm [19]. Position alone is not a useful data source, requiring environmental information to provide meaning and context. Environmental context is traditionally provided to computing systems in the form of geographical maps, which assign meanings to spatial regions. Inputting such maps is, however, a daunting and laborious task that scales approximately linearly with floor area.

Ubiquitous systems are often deployed on a small scale (a laboratory or a single room), making manual entry of the environmental data feasible. The larger problem is rarely addressed. Moreover, ubiquitous system administrators rarely provide high level information with which to interpret the concept of doorways and connectivity, instead providing a simple series of object perimeter segments. Similarly, there is no provision for synchronizing the environment model in dynamic environments, where objects are free to move.

In this paper, we apply some of the techniques learnt in the field of robotics to aid in the autonomous modelling of the environment. Robots often model their environment autonomously in order to navigate, and approaches have been developed to do so. Our aim is to use some of the vast quantities of information that a ubiquitous positioning system provides to make basic, but reliable, inferences about the world. We wish to produce techniques that can integrate with similar algorithms [10] and environments [5,13] to aid in the creation and maintenance of the world model of a context-aware system.

We will provide an overview of environment modelling in robotics, then examine why we cannot directly apply the ideas to people, and finally present an approach that allows us to indirectly use these ideas. We present real results from data captured using the ultrasound-based Bat location system [22], although we assume only a generic two-dimensional positioning system is available throughout. We briefly detail some applications of systems that can autonomously generate their world models.

2. Autonomous Navigation

Map generation is not a new subject in the engineering community. Mobile robots are often equipped with sensors to collect environmental information, which is used to construct and maintain a functional map. This map can then be used to plot an unobstructed course to a given destination. Such *autonomous navigation* dates back to the mid-1980s, and remains an active research area in robotics [2, 4, 6, 11, 12, 17, 21].

Robotics research has identified two important approaches to modelling a robot's environment: *metric* (grid-based) maps and *topological* maps. Metric maps divide the area of interest into a regularly-spaced grid of cells and associate a property inferred from sensor input with each cell. The simplest approach is an *occupancy* grid, where a cell

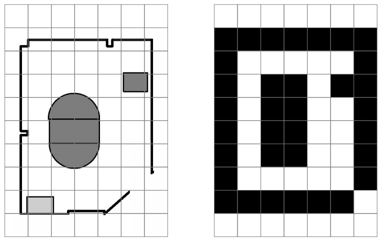


Figure 1. An example of a metric occupation grid (right) for a room (left)

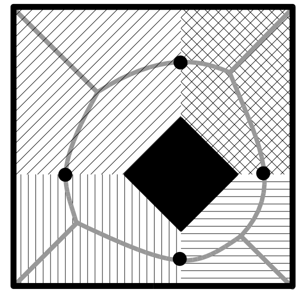


Figure 3. An example VD (sketch).

is assigned a state from the binary set {occupied, unoccupied}, according to whether an impenetrable object has been sensed within its bounds (occupied) or not (unoccupied) [14, 21]. Figure 1 provides an example.

Metric maps are easy to create and maintain, they inherently incorporate geometry, and they have a natural computational representation. However, they have large storage requirements, proportional to the square of the grid resolution, do not easily deal with dynamic environments, and can be difficult to use when path planning.

Topological maps do not attempt to represent the environment geometry directly, but instead use connectivity graphs [21]. Landmarks and objects are represented by nodes, and are graphically linked to show the existence of direct paths between them (see Figure 2). Such a map is well suited to symbolic path planners, has low storage requirements, and does not require an accurate localization of the robot. However, construction is typically complicated and recognition of position is rarely reliable.

Thrun has given a detailed survey of the two approaches, and suggested ways of combining their respective strengths to reduce their individual weaknesses [21]. The approaches described permit the use of dynamically generated metric maps to generate topological maps. Thus both map types can be easily created and maintained, and importantly kept synchronous.

In robotics, the process of generating metric and topological maps begins with instruction to the robot to follow a simple path, trying to follow the edges of the objects it observes (and thereby moving around the room perimeter). Ultrasound-based ranging sensors are used to determine the distance and direction of obstacles from the current robot position (which must be known with reasonable accuracy). When a ranging measure determines that an object lies within a particular grid cell, it alters its value to reflect the increased probability of occupation.

When a room loop has been completed, the robot applies a user-defined probability threshold to the occupancy grid in order to assign a binary occupation state to each cell, thereby forming the current metric map. To create a topological map, we can generate a *Voronoi Diagram*[1] (VD) [8, 21] from this map. Such a diagram is the locus of points with two or more equidistant obstructions. As such, it represents the paths that remain as far from any obstacle as possible. Applying the VD to the metric map produces a skeleton of the room and its landmark objects.

Once formed, we can use the VD to identify points for which the distances to the nearest equidistant objects are a local minimum. These are *critical points* [21], at which it is natural to partition the environment. Figure 3 shows an example VD. Non-circular black areas indicate impenetrable obstacles, the thick grey line represents the VD, and the black circles are critical points. The region has been partitioned into four areas (hatched) using these critical points. Each partition is then represented by a topological node, linked using the VD to form a topological map.

3. The Transition from Robots to People

The robotic navigation methodology requires a method to determine cell occupancy state from a given position, and to control the motion of the mobile automaton to ensure even coverage of the environment. For context-aware systems, we use a positioning system embedded in the environment, and the motions of people over time to create an occupancy grid equivalent, and then a topological map.

We have created occupancy grids by simply monitoring where people go. Upon application of the robotics methodology, however, this simple problem mapping failed, primarily because the system has no input or control over the motions of the people it tracks. In robotics, a single robot is programmed to move at a constant speed and in such a manner as to cover the area with an approximately uniform sighting distribution. Thus, robotic occupancy grids typically have a characteristic number of sightings of each environment obstacle, simplifying threshold determination.

[1]Readers may prefer to use a Generalized Voronoi Diagram (GVD), which forms a VD from polygonal segments rather than points. For the purposes outlined in this paper, either approach can be used, and herein we refer only to VDs.

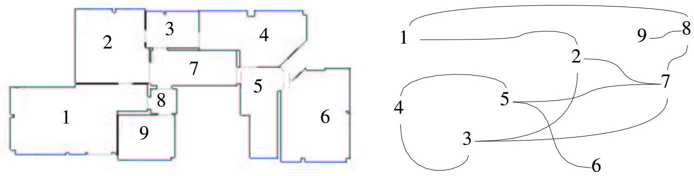


Figure 2. A floor layout (left) and its corresponding topological map (right)

People, however, move with purposes that current context-aware systems cannot define nor predict. They rarely move through an environment along its perimeter and do not aim to cover as much of the room area as possible. They move with non-uniform speeds, along repeated paths, and do not uniformly use all accessible space.

The Bat system is a three-dimensional positioning system that is installed over an area of approximately 130m^2 at our laboratory [1, 22]. It can track personnel in a three-dimensional world model to an accuracy of order 3cm 95% of the time. Figure 4(a) shows a typical two-dimensional personnel occupancy grid for a period of three weeks, extracted from Bat system data. The shading map is a linear scale from black (maximum) to white (zero). The apparent lack of information demonstrates the extreme non-uniformity of sightings which results in office spaces. Selecting an occupancy threshold to remove the (low probability) erroneous positions, whilst retaining the meaningful measures, is non-trivial. Even on a logarithmically scaled shading map as shown in Figure 4(b), non-uniformity hinders determination of a reliable threshold. Furthermore, all high resolution ubiquitous positioning systems thus far demonstrated are based on technologies susceptible to ‘dead spots,’ where sensor coverage is insufficient to accurately compute position. Such regions are unfortunately common near key areas such as doorways or walls.

4. The Linkage Diagram (LD)

We propose that ubiquitous systems record not the sightings within individual cells, but rather the *transitions* made between them, forming a *Linkage Diagram* (LD). An arbitrary set of directions is assigned to each cell (for simplicity, we will use four orthogonal directions in this paper), and each cell maintains a count of the votes it receives for a transition to the neighbouring cell in each direction (connections are always symmetric between cells). Over time, the votes build up for links, and we can use a threshold to establish the existence of a given link. By casting votes for cell linkages between sighting positions, we provide data within dead spots. We assume that the positioning system is not sufficiently reliable to *guarantee* a sighting is correct, since no such system has been demonstrated. The Bat system update rate, for example, can be significantly affected if the path from transmitter to receiver is obstructed.

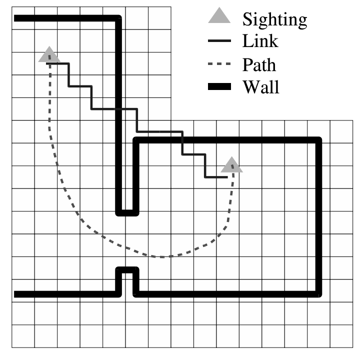


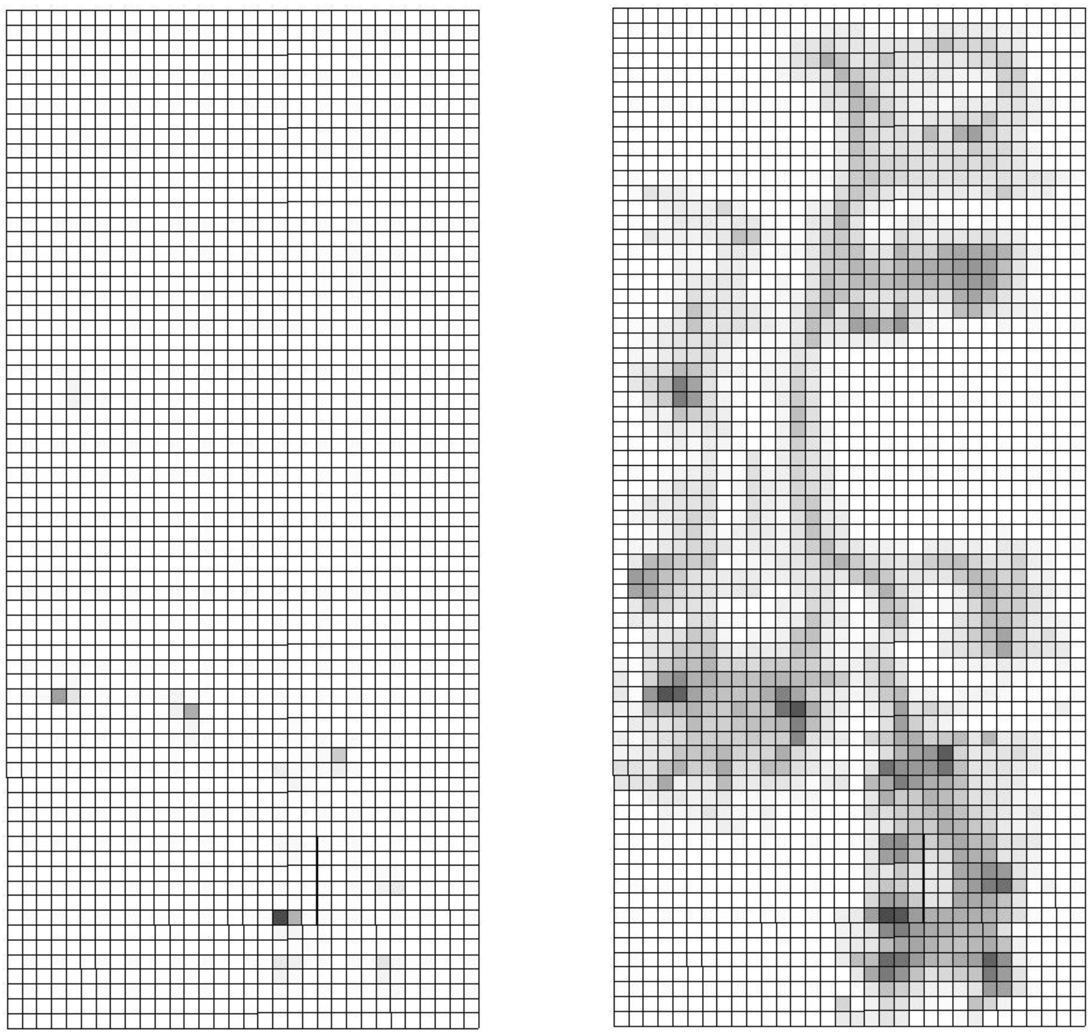
Figure 5. An example scenario.

We have found this simple approach to suffer from two major problems in real environments. Firstly, users may unintentionally obstruct the positioning medium. This can result in consecutive sightings that do not correctly represent the path taken. For example, in Figure 5, a user passes from one room to another whilst hindering positioning. The resultant linkage of cells passes directly through a wall.

The second problem relates to office space usage. The inherently non-uniform sighting distributions of Figure 4 can easily skew the cell votes, resulting in very high densities of linkage votes near desks, presenting a threshold problem.

We have addressed these problems by introducing two parameters to the LD: the *Linkage Time* (LT) and the *Linkage Length* (LL). The LT is used to significantly reduce the problem of positioning hindrance. To increment the linkage votes for the cells involved in the transition from one sighting to another, we demand that the two sightings be temporally spaced by less than LT. The value of LT is chosen to ensure that users cannot travel great distances (relative to their environment) within one LT interval. We have found a value of 5 seconds to be appropriate for our laboratory.

The LL reduces the problem of people spending disproportionate amounts of time at their desks. We require any two sightings to be spatially separated by a distance greater than LL. Selecting LL to be a length characteristic of the



(a) Linear Mapping

(b) Logarithmic Mapping

Figure 4. Occupancy grids for our laboratory.

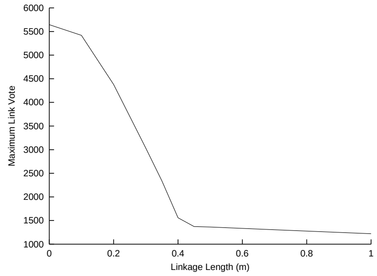


Figure 6. Variation of maximum link vote with Linkage Length for a constant cell size of 0.3m, and LT=5.0s.

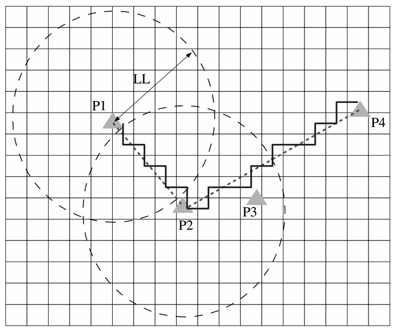


Figure 7. Constructing a Linkage Diagram.

positional variation of a seated user, we eliminate the build-up of link votes near desks. Figure 6 shows the value of the maximum linkage vote for various values of LL for our laboratory data. We see a significant drop in votes as LL increases from zero. Beyond approximately 0.4m the maximum vote is seen to settle, indicating an optimal value of LL=0.4m.

As an example of the complete method, consider Figure 7; a user moves from point P1 to point P4, and is also sighted at the interim points P2 and P3. Link votes between cells are shown with a heavy line. In the linkage diagram, P1 is linked to P2 because their separation exceeds LL. This is not true of P2 and P3, where no link is made. P2 and P4 are linked because the sighting at the latter occurs within LT seconds of that at the former. If more than LT seconds had elapsed, we could link P3 to P4 instead, forming only a disjoint link between P1 and P4. We continue in this manner, building up a large scale linkage map. The final result is a LD without the gross non-uniformity of the direct occupancy approach.

Thresholding the LD to assign binary linkage states can be achieved by discarding a certain percentage of the lowest valued links. The best percentage is a characteristic of the positioning system and its susceptibility to significant error. We obtain very reliable results using the Bat system by assigning the 'linked' state to the highest 60% of link votes.

5. Utilizing the LD

Figure 8 shows the LD for a 27 day period within our laboratory, following application of the 60% threshold. The centre of each individual area is readily apparent, and the connecting links are also observable.

We can use the diagram in a native form to generate accurate paths between two cells. Given both start and end cells, a straightforward application of Dijkstra's algorithm [7] will yield the shortest route. Even this simple application may have uses in the ubiquitous environments of the future: a visitor to a large building that implements a system with high level environmental information can query it for directions to a particular office, and be shown the best route graphically. He may even be tracked by the system and warned if he strays noticeably from the course set.

Ubiquitous environments may implement mobile automatons themselves, and thus require the capability to direct them from area to area. In such a case, the robot would wish to remain as far from obstacles as possible; any path formed using Dijkstra's algorithm will minimize the total distance by choosing pathways which *glance* obstacles. To improve on this, we can get an idea of the building 'backbone,' avoiding obstacles at the extremities of connected areas by forming a VD based on the perimeter of the connected cells. This is similar to an occupancy grid created by asserting unlinked cells to be occupied. Figure 9 shows the VD for the LD of Figure 8. We can calculate the shortest path from any point to the backbone, and then travel along the backbone to the destination [21].

Clearly, in large sensor net deployments, we expect the VD to contain a vast number of nodes, and generating routes between nodes may be computationally demanding. To reduce this load, we can derive more general topological information from the VD through a series of *pruning* and *refining* operations. We *prune* a VD by removing all vertices that connect to only one other (i.e. end points or spurs). We *refine* it by replacing all the vertices with exactly two connections, called 'conduit vertices,' with a single connection. In practice, we have found it best to refine the VD, prune it, and finally refine once more to remove any newly created conduit vertices. Figure 10 shows the result of applying these operations to the VD shown in Figure 9. The node density has been drastically reduced, and path planning between nodes can be completed much faster. Diagrams in this form help to optimise the processing of high level queries such as "can I get to Room X from here?" They fundamentally separate the connectivity from the geometry. If more detailed geographic pathways are required (to direct automatons around objects, for example), we can

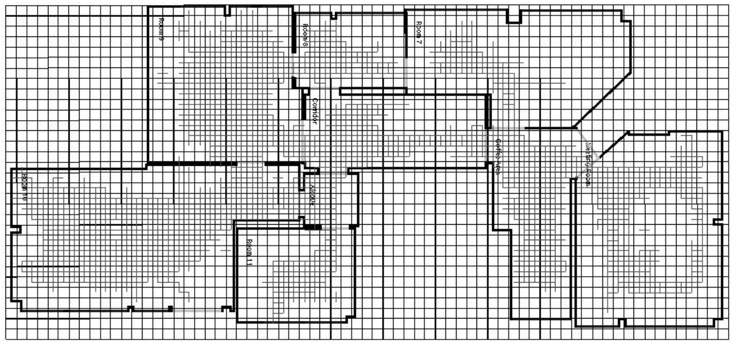


Figure 8. The Linkage Diagram formed over 27 days in our laboratory.

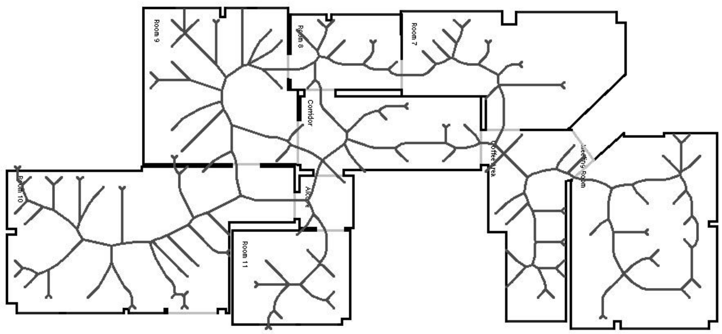


Figure 9. The Voronoi Diagram extracted from linked cells in Figure 8.

use the true VD to plot paths between the nodes identified as being necessary to traverse.

The ideal topological nodes for a topological map are the rooms, defined as the regions separated by doorways. Using only an occupation grid approach we cannot reliably find the doorway positions, even with a human interpreter. However, critical points can identify a series of candidate doorway positions. Recall from Figure 3 that critical points mark bottleneck points in the occupancy grid. We expect doorways to correspond to a subset of critical points with a clearance from the nearest obstacle to be less than or equal to half of the door width. Figure 11 illustrates the regions created by partitioning our laboratory data at critical point positions with clearances less than or equal to 0.75m. We find that doorways in the environment can be associated with boundaries between particular regions.

Identifying the correct doorway positions from the candidates and assigning the necessary semantics (room name for example), is not possible from basic positioning data. Further data from complementary sensor systems (possibly mounted on an automaton) could potentially be used to reliably combine regions to form the human notion of rooms, but sensor fusion is a requirement to assign the semantics. Without further fusion, the best approach to generating maps is to present the options to human operators, who can assign semantics and confirm and merge the correct partitions.

Once constructed, the VD has many other potential applications in environment discovery. In particular, clues to large scale static objects, especially tables, can be found. Figure 12(a) shows the pre-threshold occupancy grid for our laboratory meeting room, using the logarithmic colour map of Figure 4(b). We observe a central depression in sightings, indicative of an obstacle, but it is not possible to reliably suggest the size or shape of the object. As Figure 12(b) illustrates, the object is a standard table. It is obscured in the occupancy grid because users have, at various times, leant over it. The LD for this room thus has links which connect the outer edge of the table to regions well within its bounds. The VD shown in Figure 12(b) also exhibits this characteristic, and we find that we can use it to estimate the existence, size and shape of the object.

Furthermore, given a three-dimensional positioning system, it is possible to use cells along the VD with a lower average sighting height to infer that users seat themselves around the object, and thus suggest that the object is indeed a table. If available, accurate and reliable orientation information is also a rich source of information that can complement the VD and provide useful data on object usage.

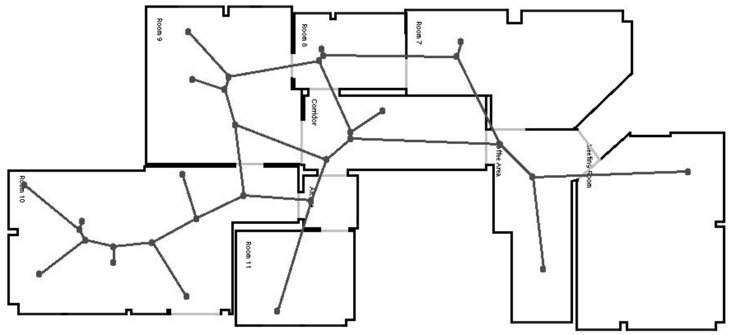


Figure 10. The pruned and refined Voronoi Diagram

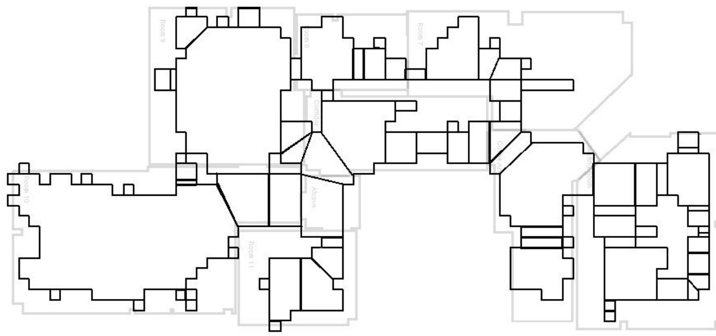
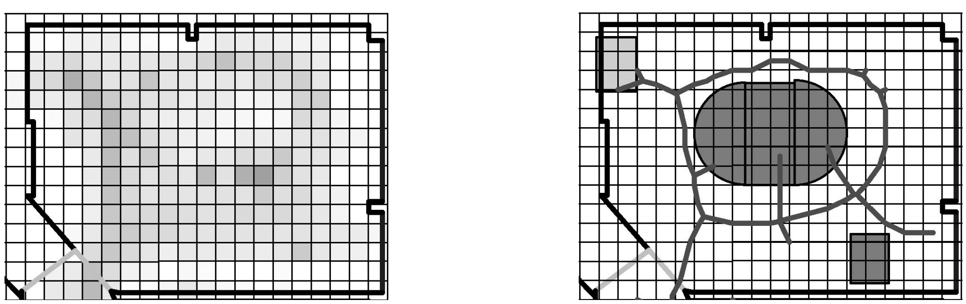


Figure 11. The result of partitioning the environment at critical points with clearances less than or equal to 0.75m.



(a) Occupancy Grid

(b) VD Pathway

Figure 12. The laboratory meeting room.

6. Applications

Autonomous configuration of world models is important for any potentially ubiquitous system. Implementation of the methods described herein can aid in the generation of topological maps for a space, and potentially characterise its usage. This is particularly useful when detailed models are unavailable by other means (CAD drawings, for example).

The methods also offer a source of information about the environment and the objects within it. Fused with further sensor data, the information should be sufficient to aid in adaptation of world models in dynamic environments. We intend to examine ways in which the algorithms can be extended to cope with the dynamic case.

There are also potential applications in outdoor positioning. Here, people again tend to be in one of two states: travelling or settled. We can extend the ideas presented to identify, connect, and characterise regions of interest [3].

7. Conclusions

Ubiquitous positioning systems are being widely deployed for use in context-aware computing environments. To use the data generated we require methods of inputting and maintaining environmental information in an autonomous manner. We have presented a novel approach to the problem of autonomously deriving geographic information using only archived personnel position data. We have shown that the approach can produce reliable topological maps. This information is useful in itself, and will complement other information sources in a sensor fusion system.

Acknowledgments

The authors would like to thank both the EPSRC and AT&T Laboratories Cambridge Ltd. for their support.

References

[1] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. *IEEE Computer*, 34(8), August 2001.

[2] G. C. Anousaki and K. J. Kyriakopoulos. Simultaneous localization and map builing for mobile robot navigation. *IEEE Robotics and Automation Magazine*, 6(3):42–53, September 1999.

[3] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with gps. In *International Symposium on Wearable Computing*, Seattle, WA, Oct. 2002.

[4] H. Asoh, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Itou, T. Kurite, T. Matsui, N. Vlassis, R. Bunschoten, and B. Krose. Jijo2: An office robot that communicates and learns. *IEEE Intelligent Systems*, 16(5):46–55, October 2001.

[5] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. A. Shafer. Easyliving: Technologies for intelligent environments. In *HUC*, pages 12–29, 2000.

[6] J. M. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. E. Schneider, J. Strikos, and S. Thrun. The mobile robot rhino. *AI Magazine*, 16(2):31–38, 1995.

[7] E. W. Dijkstra. A Note on Two Problems in Connexion with graphs. *Numerische Math.*, 1:269–271, 1959.

[8] F. Aurenhammer. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys (CSUR)*, 23(3), 1991.

[9] I. Getting. The Global Positioning System. *IEEE Spectrum*, 30(12):36–47, December 1993.

[10] R. K. Harle, A. Ward, and A. Hopper. A Novel Method for Discovering Reflective Surfaces Using Indoor Positioning Systems. To appear in Mobisys 2003, 2003.

[11] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelfth National Conference om Artificial Intelligence*. AAAI Press/MIT Press, July 1994.

[12] B. Kuipers and Y. T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 1991.

[13] Michael Coen. Design Principles for Intelligent Environments. In *Proceedings of AAAI'98*. Madison WI, 1998.

[14] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9:61–74, 1988.

[15] N. B. Priyantha and A. Chakraborty and H. Balakrishnan. The Cricket Location-Support System. *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing Networking*, August 2000.

[16] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proceedings of GLOBECOM 2001, San Antonio*, November 2001.

[17] D. Pagac, E. M. Nebot, and H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. *IEEE Trans. on Robotics and Automation*, 14(4), August 1998.

[18] C. Randell and H. Muller. Low cost indoor positioning system. In G. D. Abowd, editor, *Ubicomp 2001: Ubiquitous Computing*, pages 42–48. Springer-Verlag, September 2001.

[19] B. Schilit, N.Adams, and R. Want. Context-aware computing applications. In *Proceedings of the workshop on mobile computing systems and applications*, December 1994.

[20] K. Siwiak. Ultra-Wide Band Radio: A new PAN and positioning technology. *IEEE Vehicular Technology Society News*, 49(1), 2002.

[21] S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[22] A. M. R. Ward. *Sensor-driven Computing*. PhD thesis, Cambridge University, August 1998.

[23] J. Werb and C. Lanzl. Designing a positioning system for finding things and people. *IEEE Spectrum*, 35(9):71–78, September 1998.