Building World Models by Ray-tracing Within Ceiling-Mounted Positioning Systems

Robert K. Harle and Andy Hopper

Laboratory for Communication Engineering
University of Cambridge
[rkh23, hopper]@eng.cam.ac.uk

Abstract. Context-aware computing in location-aware environments demands the combination of real world position with a computational world model to infer context. We present a novel approach to building world models using signals inherent in positioning systems, building on the work of the robotics research field.
We implement the approach using the Bat ultrasonic location system. We observe excellent results when trying to extract the height and shape of horizontal surfaces, and demonstrate how to image and characterise object volumes.
Results are collected using personnel Bats and by using an autonomous vehicle which moves randomly. In both cases, the results are accurate and reliable.

1 Introduction

The context-aware computing paradigm [14] has sparked an interest in the development of technologies that realise the applications it offers. Thus far, the major enabling technology has emerged as location (both through absolute positioning and spatial containment), with many research systems already demonstrated to locate personnel and objects [4, 7, 10, 11, 13, 15, 17, 18], and many more on the horizon.

Context-aware systems must integrate location information with a knowledge of the world in which they operate. This knowledge is contained within a world model, and may be as detailed or sparse as the applications which utilise it demand. As a minimum, the authors have found by experience that a useful system should be aware of room bounds, computer host positions, and the location of major office furniture. These facilitate the majority of applications such a hot-desking and "follow-me" applications.

The experiences of the authors with the SPIRIT context-aware system [1], have highlighted a series of practical problems that regularly manifest in a real world deployment. Objects modelled by SPIRIT, but not explicitly tracked by sensor systems, are continually observed to unexpectedly move, reconfigure, or be removed altogether. As an example, users commonly shift and re-orient their desktop display.

The resultant loss of synchronisation between the real world and the world model can be problematic for context-aware systems. If a monitor is moved, for example, the ability to hot-desk to it whilst in its vicinity is immediately lost. The system continues to make decisions consistent with its model, but now inconsistent with the real world. Users become confused, and start to distrust the system, reducing its value.

To prevent this, context-aware systems must be able to respond to changes in the real world. Current implementations use a world model that is manually configured at initial switch on. Adaptation to any subsequent changes in the real world relies on human administrators observing those inconsistencies and taking the time to correct them. On the scale of a laboratory implementation, this is adequate. However, larger deployments clearly cannot depend on this approach.

This paper presents a method of using signals within positioning systems to infer the presence, position, and shape of objects in the environment with minimal human involvement.

2 Robotics and Positioning Signal Propagation

Unobtrusive positioning systems rely on the propagation of wave phenomena. The characteristics of the propagation (such as time-of-flight) or the wave itself (phase) provide information that can be combined with the information from other waves to compute a position. For example, the Bat system [17] uses the time-of-flight measurements of ultrasonic waves propagating between a transmitter and multiple, ceiling-mounted, receivers to perform a multilateration calculation and estimate the transmitter position.

Given the position of a transmitter and a receiver that detects an emitted signal, we establish a vector between these positions, known as a *ray*. Herein, it will be assumed that the positioning system uses a mobile transmitter and fixed receivers. All concepts are directly applicable to the inverse situation by symmetry arguments. As the transmitter is moved around, a series of new rays are established with each successful positioning calculation.

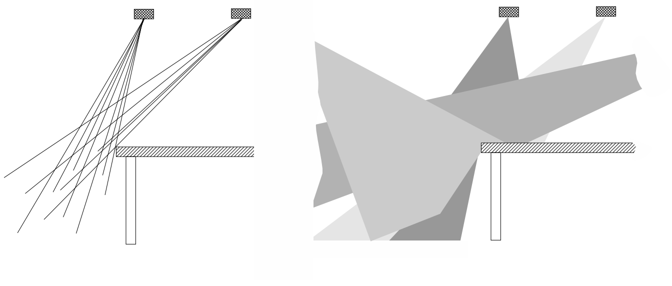


Fig. 1. The build up of rays around a table

Over time, the rays penetrate into three-dimensional space. Where objects made of material that is opaque to the positioning medium exist, no rays are expected (illustrated in Fig. 1). The essential premise, then, is to facilitate the aggregation up of rays within an environment, and provide an analysis algorithm to extract information from the regions that no rays intercept.

In a real system, rays may build up very quickly. A three-dimensional position requires a minimum of three receiver sightings, and hence provides at least three rays. At a conservative update rate of 0.1Hz, we generate a minimum of 25,920 rays per person per day. Robust systems are likely to vastly exceed this minimum. It is therefore not storage efficient to store the details of every ray within the system.

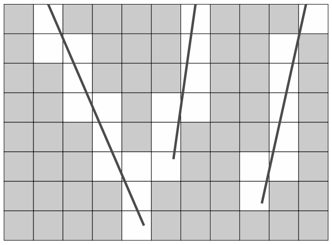


Fig. 2. A vertical slice through an occupancy grid. Grey cells are occupied. Rays are represented by thick dark lines.

Instead, we can use a three-dimensional *occupancy grid*. Occupancy grids were first proposed by Moravec and Elfes in the field of robotics and autonomous navigation, where robots must use attached sensors to model their environment. There exists extensive literature in the field pertaining to mapping and exploration [2, 3, 8, 9, 16].

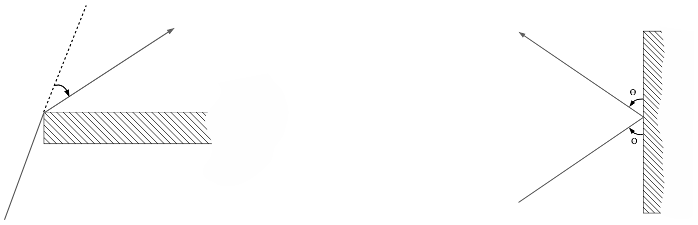
For the purposes of this paper, an occupancy grid is a three dimensional construction that segments the space of interest into regular cubes. Each cube is associated with a binary state from the set {occupied, unoccupied}.

A ray is quantised onto a grid by determining the cells it intercepts, and assigning them the unoccupied state (Fig. 2). Using a spatial grid reduces the storage requirements, and eases analysis. The accuracy of such a representation depends on the choice of cell size. A smaller cell size necessitates larger storage requirements, but potentially gives increased accuracy.

3 Real World Difficulties

Ideally a ray-tracing system would record the receivers that receive any positioning pulse and store the corresponding rays in some form. However, to do so would make the implicit assumption that the path traversed by the positioning signal was straight and direct between the transmitter and receiver. In reality, waves are subject to a series of physical phenomena which may cause deviations from this ideal. In particular, waves may be diffracted (Fig. 3(a)) or reflected (Fig. 3(b)).

The possibility of signals taking non-direct paths has traditionally been a serious problem for positioning systems. It introduces a source of significant error into the positioning algorithm. In any set of positioning signals, we wish to establish those that traversed direct paths, and use *only* those when ray-tracing.



(a) Edge diffraction

(b) Specular reflection

Fig. 3. Wave phenomena causing signal path deviations

We can reliably estimate the direct-path subset of measurements by discarding the rays for all measurements that are discarded by the positioning algorithm. In this way we maintain all measurements that are consistent with the returned position. By ensuring that the positioning algorithm is resilient to non-direct measurements, these measurements most likely represent the direct-path subset.

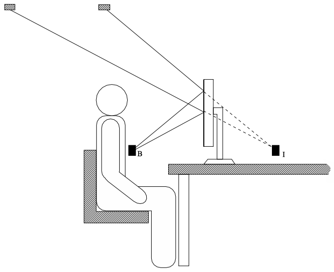


Fig. 4. Reflection images in the Bat system

However, in extreme cases, it is still possible to misidentify direct signals, as illustrated in Fig. 4. Here, *no* signals propagate to any receivers without first specularly reflecting from a monitor. The resultant time-of-flight (or equivalent) measurements are consistent with a position reflected through the plane of the monitor, and a positioning algorithm will converge on this reflected position, producing rays that obscure the monitor.

The positioning algorithm implemented within the Bat system uses a non-linear model of the data. The standard usage of such an algorithm is to create a model from all measures, discard the largest outlier and repeat until a certain error level is reached, or the algorithm diverges. As described in [6], the Bat algorithm modifies this technique, and discards only the measures with the largest *positive* residuals, thereby encapsulating the idea that an ultrasonic signal cannot travel faster than the speed of sound. The effect of this is that a single direct measurement is enough to prevent the positioning algorithm converging. Thus, situations where a reflected position is returned can be minimised, but not eliminated.

To handle the introduction of such error requires a more detailed approach when forming an occupancy grid. In robotics, a probability of occupancy is assigned to each cell rather than a binary state, and a final probability threshold applied to convert to a binary grid [16]. This works well because the error model for the sensor is easily modelled, allowing for meaningful probabilities. In this methodology, a ray becomes a probabilistic beam with reduced probability of occupancy further from the central axis.

With a positioning system, this methodology can be difficult to apply. Inherent positional error can be used to create a beam, using a Gaussian distribution centred on the position estimate, with a width determined by any available error estimate. However, this does not account for reflections, which are not predictable and hence not reliably modelled. Similarly, if the grid cell size exceeds the typical error for a position, quantising beams onto the grid is little different from the quantisation of rays, but more demanding in both computation and storage

An alternative approach when the typical positional error is of the same order as, or less than, the cell size is to reduce the effect of erroneous measures statistically. A *voting grid* can be formed, whereby each cell has an associated voting count rather than a binary occupancy state or a floating point probability. This voting count is incremented whenever a ray intercepts the cell.

When reflected positions such as that shown in Fig. 4 are possible, we expect to see a build up of votes within those cells in proportion to the probability of finding the incorrect reflected position. The positioning algorithm described in [6] makes this probability sufficiently small that we can identify the erroneous cell interceptions by thresholding the voting grid to form a binary occupancy grid.

4 Ceiling Mounted Positioning Systems

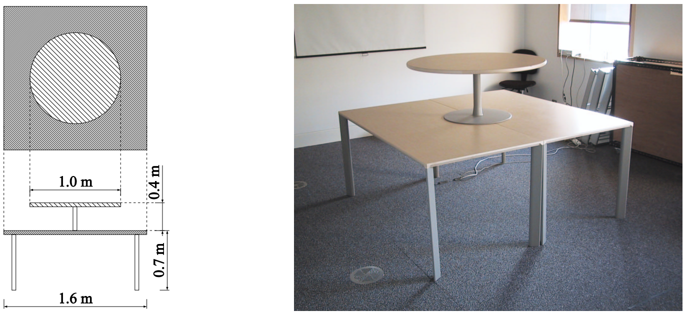
The systems that have exhibited the highest positioning accuracy to date are primarily ceiling mounted [10, 12, 13, 17]. The Bat system, for example, uses a mobile ultrasonic transmitter and a series of ceiling mounted receivers to position to within 3cm (95% confidence level).

Such systems are well suited to ray-tracing because they typically have a reasonably large density of receivers and produce accurate positions, ensuring a fast and extensive build up of reliable rays. However, the geometry of the situation means that volumes vertically below a surface (such as the space underneath a table) cannot be mapped. So such systems are best restricted to the determination of specific surfaces.

4.1 Horizontal Surfaces

Determining the height, size, and shape of horizontal surfaces in the human environment is particularly useful. These are the surfaces upon which we can work, and upon which we store items of interest to context-aware systems.

To demonstrate ray-tracing, an experiment was performed using a combination of two tables within a room covered by the Bat system. The two tables were chosen to be of differing heights and shapes, and were setup as shown in Fig. 5.



(a) (b)

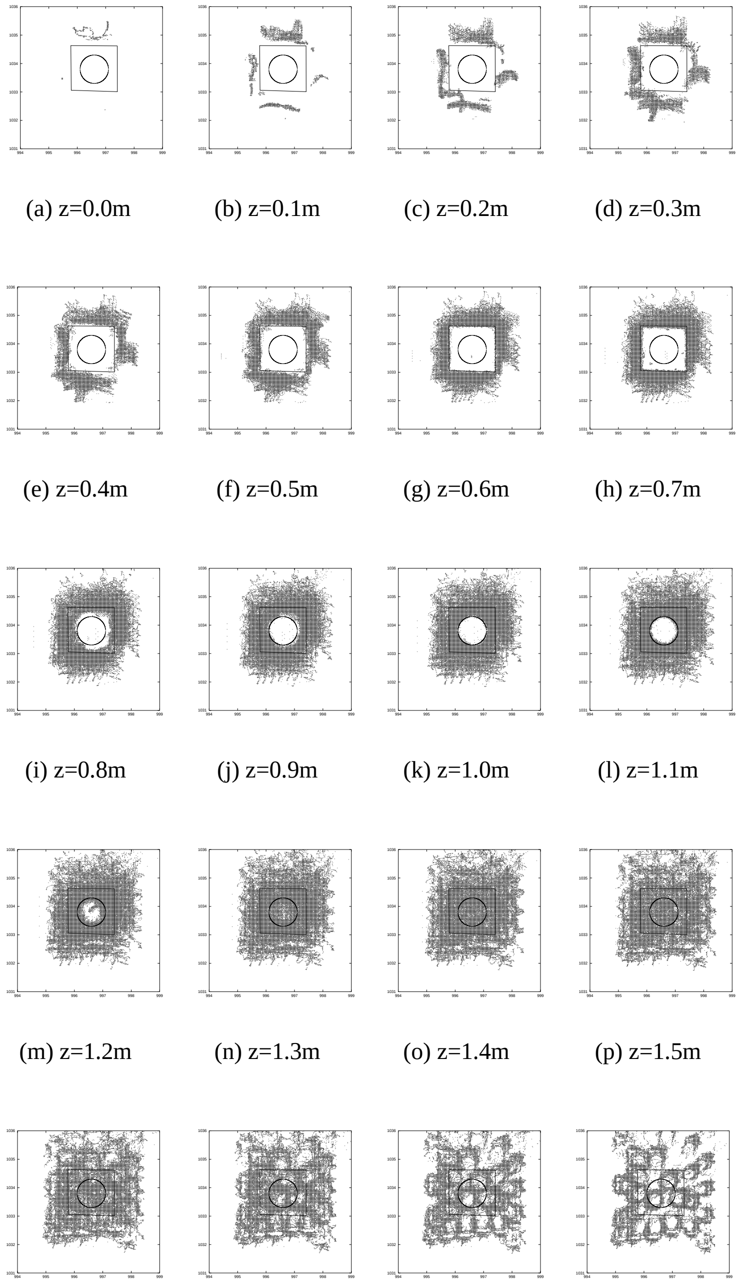
Fig. 5. Table configuration for tests

A Bat was moved at random in and around the vicinity of the tables for a period of a few minutes, whilst the Bat system was held in a high update rate to maximise result collection. A total of 6,630 positions were recorded, giving rise to 77,631 rays. The rays were quantised onto a voting grid with dimensions 3.0mx4.0mx2.2m and a cell size of 0.02m. A low threshold was applied, since there were no near-vertical surfaces to reflect positions, and thus very few erroneous rays. Figure 6 shows the resultant ray intersections with horizontal planes at different heights, *slices*, superimposed with the measured outlines of the tables.

At low heights, we observe the intersection distribution to be scarce and highly globular; a result of fewer sightings made at that height. Even so, we immediately see the emergence of the larger square table. The table outline becomes sharper as we approach the table height of 0.7m. Beyond this we observe the circular outline of the second table begin to form. Again, we qualitatively note that the outline is sharpest at a height of 1.1m; the correct height for the second table. We then pass through a region of extensive intersections with no large scale regions present. Above a height of approximately 1.4m, we see small near-circular slices of the individual intersection cones associated with, and centred on, each receiver position (see Fig. 6(t)).

Autonomous Extraction of Specific Surfaces Once a binary occupancy grid is established, various techniques can be used to extract object details. It is possible to apply image processing techniques such as the Sobel edge detector or the Hough transform [5]. These, however, must be performed in three-dimensional space, which is slow and cumbersome. They also produce extensive edge information, which can be difficult to reduce and amalgamate to form polygonal object representations.

It is also possible to use the characteristics of the occupancy grid at various heights to autonomously extract information about an individual surface. Given a seed point, $\mathbf{s}$, located approximately at the centre of the surface we can use region growing code on a series of slices at different heights to examine the geometrical properties of any region containing $\mathbf{s}$.



(a) z=0.0m (b) z=0.1m (c) z=0.2m (d) z=0.3m

(e) z=0.4m (f) z=0.5m (g) z=0.6m (h) z=0.7m

(i) z=0.8m (j) z=0.9m (k) z=1.0m (l) z=1.1m

(m) z=1.2m (n) z=1.3m (o) z=1.4m (p) z=1.5m

(q) z=1.6m (r) z=1.7m (s) z=1.8m (t) z=1.9m

Fig. 6.

At each height, we can form a dimensionless constant, R, for the region, where R is defined in terms of the region perimeter, P, and the region area, A

$$R = \frac{P^2}{A}. \tag{1}$$

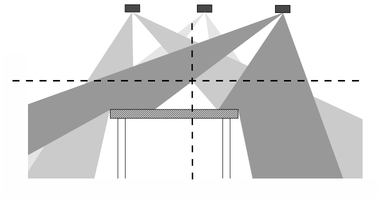


Fig. 7.

When no region exists around seed s, we stop searching, making the implicit assumption that no surfaces exist above this height. We can justify this by realising that, for there to be no region, rays from the ceiling must penetrate into the area and a significant volume above it, as illustrated in Fig. 7. The dashed horizontal line represents the first height at which no region can be found using the seed shown as a vertical dashed line. We can conclude that any object above this height would likely be ceiling mounted itself.

The quantity R provides an estimate of how noisy the region shape is. Since we expect edges to be smooth for comfort, safety and aesthetic reasons, we expect to observe a strong local minimum in a plot of R versus height, z. We term such a plot the *profile plot*.

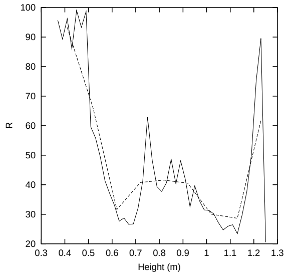


Fig. 8. The profile plot

Figure 8 shows the profile plot for the data collected for Section 4.1, using a seed manually calculated to be the centre of the true table position. The underlying distribution is noisy and can be smoothed with a simple boxcar average (dashed line in Fig. 8).

Table	Perimeter Error	Area Error	**R** Error	Height Error
Square	-0.001m (-0.02%)	+0.28m^2 (+11.2%)	+15.4	-0.013m (-1.9%)
Circular	-0.28m (-0.09%)	-0.15m^2 (-19.5%)	+16.1	-0.02m (-1.8%)

Table 1. Errors

This highlights two major minima at heights of 0.7m and 1.1m. We can then find the local minimum within this region from the original profile plot. The results shown in Fig. 8 estimate the heights of the two minima to be 0.67m and 1.07m, in good agreement with the measured heights of the table surfaces (0.683m and 1.090m, respectively).

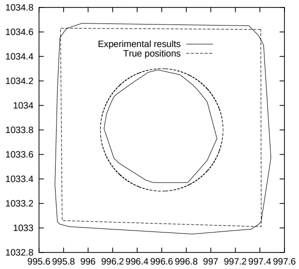


Fig. 9. Autonomous extraction results

Figure 9 shows the convex hull of the perimeter of each region extracted at the heights with minima in the profile plot. The true shapes and positions of the tables are superimposed for comparison. Table 1 details the associated errors.

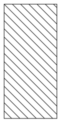
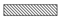
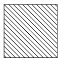
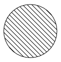
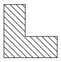
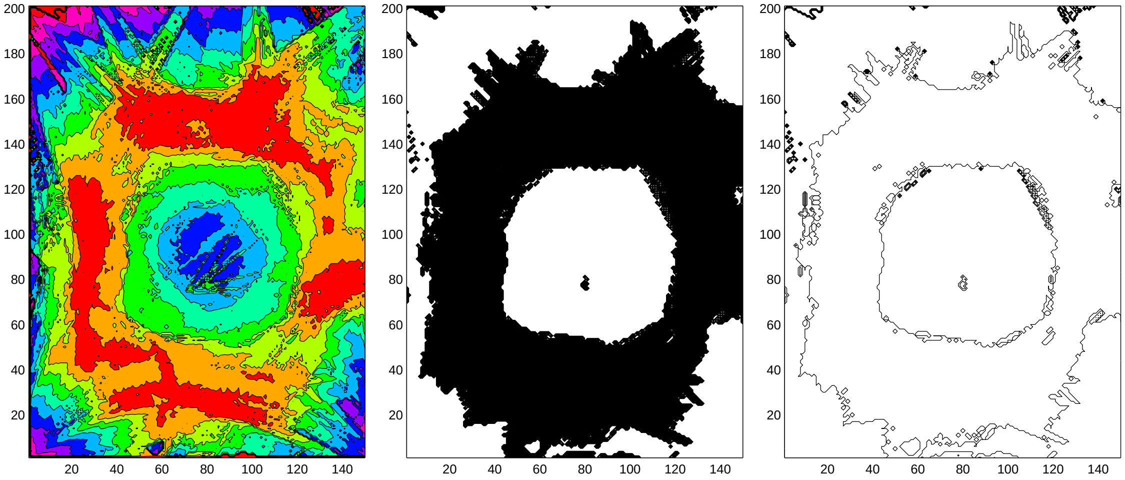
Shape					
R value	18.0	28.8	16.0	4.0	25.0

Table 2. Profile plot ratios

Note that the value of R is also indicative of shape of the object. Table 2 gives the expected values of R for a series of shapes. Whilst it has not been possible to reliably extract the shape given any experimentally determined value of R, the order of magnitude of R is a useful indicator as to whether an object is genuinely present. For example, finding $R > 50$ is unlikely for an object in an office environment.

Seed Identification Section 4.1 described a method of extracting a surface shape and height given a seed point which lies within its bounds. To autonomously extract all the surfaces in a region, then, we require a method of identifying the seed points within a data set.



(a) Contour plot

(b) Thresholded using average value

(c) Region extraction

Fig. 10. Using contour plots to determine seeds

Consider a three dimensional occupancy grid with all rays quantised into it, as described in Section 2. We can examine each vertical column of cells in turn, and create a two-dimensional grid which contains the largest number of consecutive empty cells within that column. This creates a contour plot of the number of empty cells within a column, as shown in Fig. 10(a) for the table data above.

By applying a threshold to the contour plot (Fig. 10(b)), we can use region growing algorithms to extract regions (Fig. 10(c)) from which we obtain seed points. Figure 11 illustrates that the choice of seed point within a region is important. It shows two seed points, $S1$ and $S2$, and sketches of their resultant profile plots. We observe that $S1$ prematurely ends the plot, and obscures the local minimum. To avoid this situation, we take the centre of the column with the highest count as a seed point ($S2$). This ensures that we do not prematurely cease searching in the profile plot. If multiple cells have this same maximum, it suffices to take the average centre position.

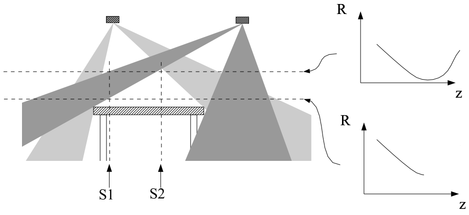


Fig. 11. The importance of seed choice

Given a set of seed points, we create a profile plot for each, and search for minima and subsequently horizontal surfaces as in Section 4.1.

4.2 Non-Horizontal Surfaces

Non-horizontal surfaces are potentially problematic for ray-tracing in ceiling mounted systems. Finding such surfaces is possible by first performing a coordinate rotation as illustrated in Fig. 12, and then applying the profile plot analysis of the previous section.

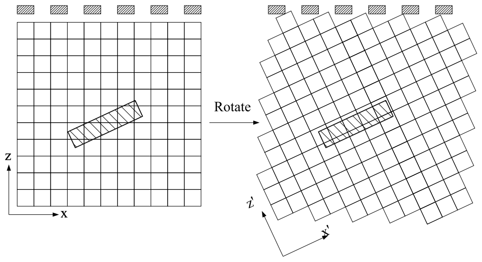


Fig. 12. Coordinate system rotation for non-horizontal surfaces

Such an approach can work well given prior knowledge of the surface inclination. If this information is not available, as is often the case, we require that a complete profile plot be created for every three-dimensional direction and the most likely plot then determined. This is computationally expensive and hence an impractical solution.

Profile plots may still exhibit useful characteristics. To demonstrate this, a large cardboard wedge was suspended within a room (Fig. 13), and 42,270 rays collected in its vicinity. In effect, the wedge is a stack of horizontal surfaces of diminishing area with increasing height. As expected, then, we see an extended minimum in the profile plot (Fig. 14), which does not lend itself to the analysis of Section 4.1.

However, it is of use to consider the variation of region area with height. A three-dimensional object will have a characteristic trace in a plot of these quantities. For

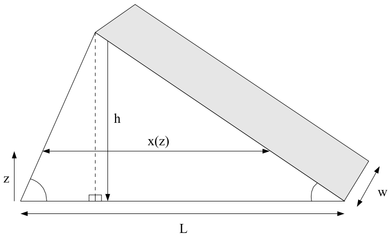


Fig. 13. Wedge example

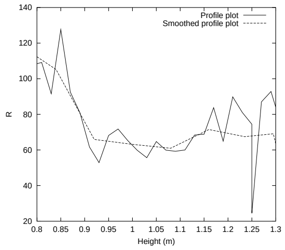


Fig. 14. Wedge experimental profile plot

example, consider the arbitrary wedge in Fig. 13. At height, z, its horizontal cross-section has area,

$$A(z) = w \cdot x(z). \tag{2}$$

Where the dimension, x, varies as

$$x(z) = L - \frac{z}{\tan\theta_1} - \frac{z}{\tan\theta_2}. \tag{3}$$

Simple geometry dictates that

$$L = \frac{h}{\tan\theta_1} + \frac{h}{\tan\theta_2}. \tag{4}$$

Combining (2), (3), and (4), then, we find that

$$\begin{aligned}\frac{dA}{dz} &= -w \cdot (\frac{1}{\tan\theta_1} + \frac{1}{\tan\theta_2}) \\ &= -\frac{w \cdot L}{h}.\end{aligned}$$

This relationship is characteristic for a general wedge. Figure 15 shows the plot of area versus height for the experimental data. From (5), and the measured parameters of the wedge, we expect to find a gradient of -1.55m. The best fit line shown in Fig. 15 has a gradient of -1.56 ± 0.01m.

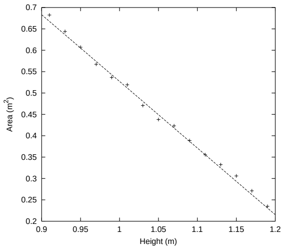


Fig. 15. Variation of area with height

Whilst it is not possible to infer the shape directly from this gradient measurement (due to ambiguities), it serves as an identifier for an object. Such a quantity can be used to determine whether a newly discovered object is truly a new object, or a previously identified object that has moved. Note that more complex objects have a correspondingly more complex variation of cross-sectional area with height.

In order to analyse new objects of which there is no prior knowledge, volumes can be imaged using the unoccupied cells within an occupancy grid. Figure 16 illustrates the unoccupied cells found in the area of the cardboard wedge. The true wedge shape and position has been overlaid to illustrate the accuracy in determining occupied volume.

In general, we have found the volume images to represent the top features of objects to a good approximation. Due to the nature of ceiling-based positioning systems, information below these features cannot be extracted.

5 Autonomous Ray Collection

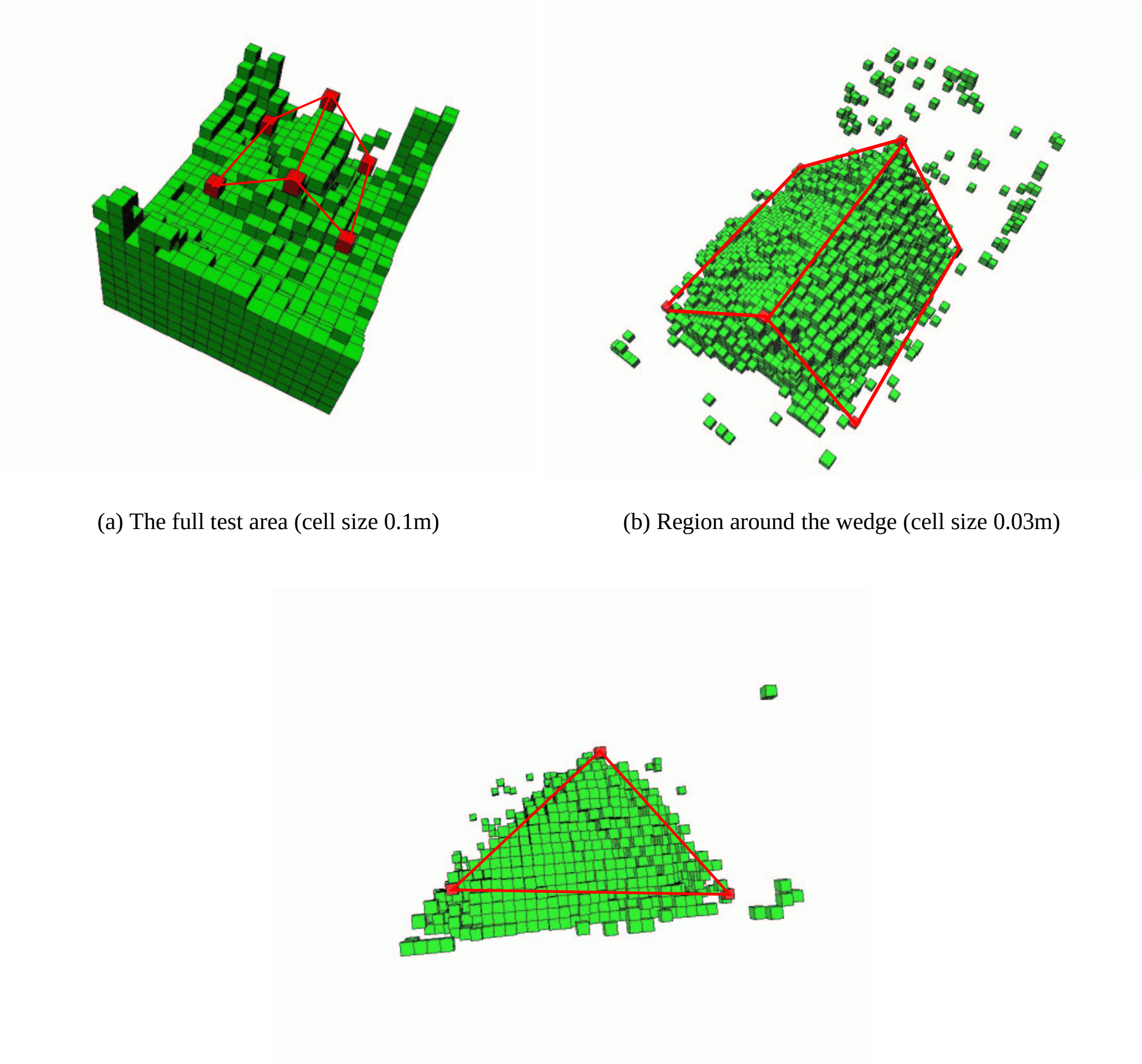
Due to the nature of the Bat system, signals are designed to propagate from Bat height (usually chest height) to the ceiling. This height range does not typically contain any objects, reducing the value of the methods described above when using only personnel movements to collect data. This limitation can be solved by distributing the receivers across a variety of heights, or by using Bats that lie below the height of the objects of interest. The latter approach can be realised with the use of a small autonomous vehicle.

Figure 17(a) shows a simple prototype of such a vehicle, programmed to move with the algorithm of Fig. 17(b). This algorithm effectively moves the vehicle along random paths within an area.

Two tables were arranged within a room as shown in Fig. 18(a). The vehicle, with a Bat attached, was allowed to move freely around the room for a period of 30 minutes. The path taken is superimposed on Fig. 18(a).

The two autonomously identified regions, along with their profile plots, are shown in Fig. 18. The primary minima estimate the table heights to be 0.73m and 0.72m. These correspond to errors of +0.05m and +0.04m, respectively. This is in excellent agreement, giving an error with the same order of magnitude as the underlying positioning system.

Note that a misleading minimum appears at a height of approximately 0.2m. This is an artifact of all results being collected at this height. Thus the slice at this height is sim-

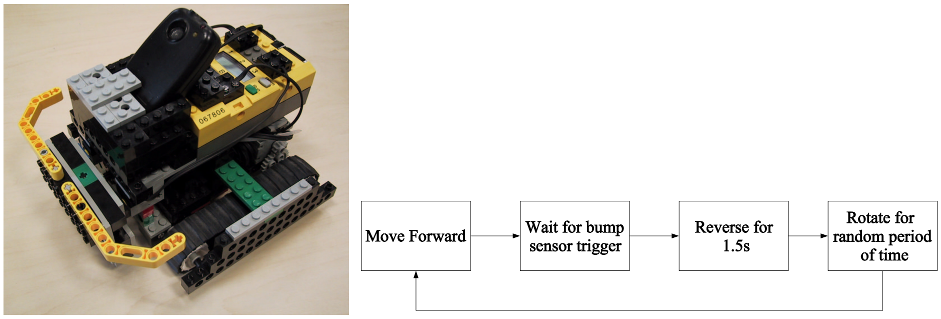


(a) The full test area (cell size 0.1m)

(b) Region around the wedge (cell size 0.03m)

(c) Region around the wedge (cell size 0.03m)

Fig. 16. Unoccupied cells


Move Forward
Wait for bump sensor trigger
Reverse for 1.5s
Rotate for random period of time

(a) Autonomous vehicle

(b) Movement Algorithm

Fig. 17.

ply the sighting distribution, which is not noisy, and therefore has a low R value. When collecting sightings at a single height, it is hence justifiable to ignore any minimum that is within a small distance of this height.

6 Limitations

The ray-tracing methodology is limited at any boundary of a ceiling mounted sensor network. In the specific case of the Bat system, ultrasonic signals cannot penetrate walls, making the boundaries rooms.

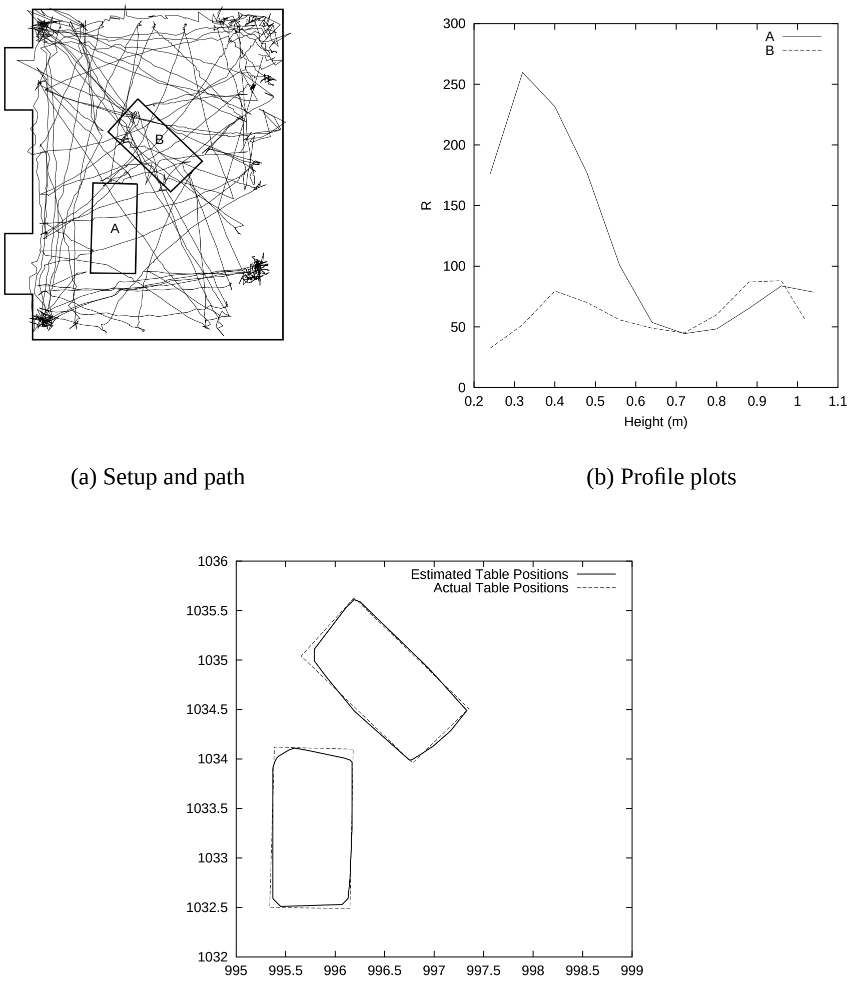
Positioning itself is limited in the same regions. Consider Fig. 19(a), a topological view of a room and its ceiling mounted receivers. Two transmitter tags, A and B, are shown in symmetrical positions, along with their corresponding areas of possible reception on the ceiling. Since we require a minimum of three measurements to distinct receivers to calculate a three-dimensional position, we see that only B is positioned due to its orientation.

It is possible to rectify the problem by increasing receiver density along the boundaries, but this involves increased installation and maintenance requirements, and does not guarantee reliable positioning (since highly clustered receivers do not give a good geometry for positioning). Even when this is done, however, there is a reduced likelihood of transmitters being near to walls.

The result is a series of regions near walls and corners where rays do not penetrate because of the asymmetrical receiver distribution, as illustrated in Fig. 19(b). Objects in such regions cannot be found using the ray tracing methods described above. Instead, it is possible to use reflected signals to get information about such areas [6].

It is useful to note that a good receiver geometry for positioning gives rise to a good geometry for ray-tracing, and thus applying the ray-tracing methodology to a positioning system that exhibits good coverage will likely yield good results.

When imaging a volume, the accuracy of the volume shape at representing sloped surfaces depends on the density and coverage extent of receivers, and the actual incli-

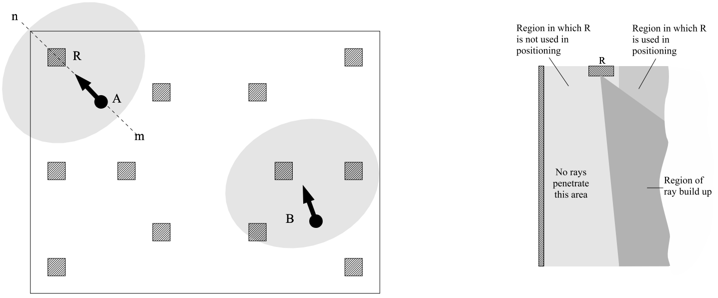


(a) Setup and path

(b) Profile plots

(c) Extraction results

Fig. 18.


n
R
A
m
B
Region in which R is not used in positioning
Region in which R is used in positioning
R
No rays penetrate this area
Region of ray build up

(a)

(b) View perpendicular to line n-m

Fig. 19. Boundary effects

nation. Since we typically aim to reduce the density in deployments, and the coverage extent is limited by room bounds, imaging may not extract a clean and accurate representation without a high result density from a diverse set of rays.

7 Conclusions and Further Work

The authors have presented a use of time-of-flight positioning system signals to map the environment. The approach permits for the recognition that an object exists within an environment, and the determination of its height and shape.

The ideas have been implemented and demonstrated to work using results from a ceiling-based ultrasonic positioning system. They should transfer directly to more generic positioning systems that do not have ceiling-mounted receivers. Such systems have fewer limitations, since rays are established in many different orientations, rather than solely toward the ceiling. It is hoped to demonstrate this using a modified Bat system that distributes the receivers throughout three-dimensional space.

This paper primarily addresses the creation of a world model from an unconfigured state. Equally, the process could be used to maintain world models in dynamic environments, although the update rate would be relatively slow, and highly dependent upon the sighting distribution. Future work will address the need for timely and reliable updates to spatial subsections of an occupancy grid.

8 Acknowledgements

The authors are grateful to both the EPSRC and AT&T for supporting this work through a research grant and CASE award respectively.

References

1. N Adly, P. Steggles, and A. Harter. SPIRIT: a resource database for mobile users. *Proceedings of ACM CHI'97 Workshop on Ubiquitous Computing, Atlanta, Georgia, March*, 1997.
2. Elfes A. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, June 1987.
3. Elfes A. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.
4. I. Getting. The Global Positioning System. *IEEE Spectrum*, 30(12):36–47, December 1993.
5. R. C Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.
6. R. K. Harle, A. Ward, and A. Hopper. A Novel Method for Discovering Reflective Surfaces Using Indoor Positioning Systems. In *Proceedings of the First Int. Conf. on Mobile Systems, Applications, and Services (Mobisys)*, 2003.
7. J. Hightower and G. Borriello. Location sensing techniques. *IEEE Computer*, August 2001.
8. H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9:61–74, 1988.
9. Moravec H. P. and Elfes A. High resolution maps from wide angle sonar. In *Proceedings of the IEEE Int. Conf. Robotics and Autmation*, 1985.
10. 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.
11. D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proceedings of GLOBECOM 2001, San Antonio*, November 2001.
12. N. B. Priyantha, K. L. M. Allen, H. Balakrishnan, and S. J. Teller. The cricket compass for context-aware mobile applications. In *Mobile Computing and Networking*, pages 1–14, 2001.
13. C. Randell and H. Muller. Low cost indoor positioning system. In Gregory D. Abowd, editor, *Ubicomp 2001: Ubiquitous Computing*, pages 42–48. Springer-Verlag, September 2001.
14. B. Schilit, N.Adams, and R. Want. Context-aware computing applications. In *Proceedings of the workshop on mobile computing systems and applications*, December 1994.
15. K. Siwiak. Ultra-Wide Band Radio: A new PAN and positioning technology. *IEEE Vehicular Technology Society News*, 49(1), 2002.
16. S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
17. A. M. R. Ward. *Sensor-driven Computing*. PhD thesis, Cambridge University, August 1998.
18. J. Werb and C. Lanzl. Designing a positioning system for finding things and people. *IEEE Spectrum*, 35(9):71–78, September 1998.