Dynamic World Models from Ray-tracing

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

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

Abstract

Context-aware computing systems demand an accurate and up-to-date world model which computationally represents the environment they oversee. Systems to date tend to have small-scale implementations with hand-programmed world models.

In real environments, manual creation and maintenance of such models is infeasible. This paper presents a novel method of using signals propagating in a multilateration positioning system to assist in creating and maintaining models of a dynamic world. It builds on a previous method for discovering objects in static environments.

The methods are implemented and evaluated using a real positioning system. They are shown to build three-dimensional occupancy grids of indoor volumes, and have the capability of modifying those grids as time proceeds and the environment is reconfigured.

1. Introduction

Context-aware computing [29] is the combination of live sensor data with a computational model of the environment. Its goal is the production of contextual information that computer systems embedded around the environment can use to react to the current situation. It is an emerging research area in the field of Ubiquitous Computing. Much of the current research has centred on location information since this provides a rich source of context. Sensing who is standing before a particular machine allows automatic reconfiguration to suit preferences; tracking people allows for new security measures; tracking objects permits for a more controlled and efficient working environment.

Whilst both location technologies and context-aware middleware continue to advance [11, 16, 24, 25, 28, 30, 33, 35], context-aware implementations suffer currently from difficulties with the *world model* - the computational representation of the environment from which any context is derived. The more accurately and completely an environment is modelled the richer the ultimate contextual information. Accurate modelling is, however, a difficult goal to achieve. Todays context-aware implementations span small-scale areas and world models can be created manually by making extensive measurements. This approach does not scale well with area size nor account for the inherently dynamic nature of the world. Changes to the model must again be input manually.

These problems have inspired an interest in methods to create and maintain world models with minimal human exertion. The authors have previously detailed a series of approaches to building useful world model information using a high precision ubiquitous positioning system [12–14]. These and other methods have been shown to assist in creating a world model of a static environment, given a sufficiently dense and spatially diverse set of positioning events. However, experience with context-aware systems [3,22] has shown that the generation of an accurate model for a static environment is only part of the problem. Maintaining it on a daily basis is a necessity. Whilst many of the discovery techniques for static environments can be applied, modification is required to cope autonomously with dynamic environments. This paper introduces a significant extension to the ray-tracing technique described in [14] (summarised in Section 3). The extension moves the method into the realm of dynamic environments.

The rest of this paper is organised as follows. Section 2 reviews related work in context-aware computing and dynamic environment modelling. Section 3 summarises the usage of ray-tracing for static object discovery. Section 4 examines the requirements for a dynamic system and the modifications necessary to the ray-tracing methods. Section 5 details the implementation of the methods using the Bat ultrasonic positioning system and documents the results of doing so.

2. Related Work

2.1. Context-aware computing

Context-aware computing is a subset of the Ubiquitous Computing vision [9, 20]. Context-aware systems use sensor inputs to derive contextual information to better interact with users.

The applications of context-aware computing have been previously described by Schilit [29] and seminal advances have been made by research labs in the last decade [2, 3, 7, 15, 22].

2.2. Fine-Grained Location systems

The *Constellation* system [10] is a positioning system designed for accurate tracking in augmented reality systems. It uses a series of ultrasonic sensors worn on the belt and head of a user. Ultrasonic transmitters are installed in the environment and periodically emit signals which can be measured by the mobile sensors. Multilateration is used to calculate position to a precision of approximately 0.005m.

The *Bat* system [33] provides accurate positioning for powered tags by multilateration of the time-of-flights of ultrasonic signals. An active mobile transmitter is worn by personnel or affixed to objects. Radio signals are used to synchronise the emission of ultrasonic pulses, which are detected by a matrix of receivers installed at known locations in the ceiling. The system is designed for tracking personnel and has a precision of approximately 0.03m.

The *Cricket* system [24, 26] uses a decentralised approach with a low density of combined ultrasonic and radio beacons distributed around the environment. Mobile receivers detect the beacon signals and compute their own location, thereby protecting user privacy. The positional accuracy exhibits a high variance.

The *HiBall* system [34] uses an array of LEDs installed in the ceiling. Users wear a device augmented with infrared photo-diodes and the LEDs are flashed in such a manner as to permit positioning.

The *EasyLiving* project at Microsoft uses video cameras to track a small number of users around an environment [7]. Microsoft also developed the RADAR system [5, 6] which uses radio propagation in environments equipped with wireless LAN systems to track mobile devices. RADAR has since inspired a number of location systems based on wireless LAN [1, 31, 36], but none exhibit the same accuracy properties of current ultrasonic systems.

Other positioning systems are on the horizon. In particular, ultra-wideband radio systems show promise but the technology remains in its infancy and few positioning results are available for a typical indoor environment [19]. The best indoor positioning systems to date are based on the multilateration of signal times-of-flight.

2.3. Autonomous Navigation

The construction and maintenance of a world model has traditionally been the interest of autonomous vehicles - mobile robots must learn about their environment in order to navigate around it. They typically use a variety of sensors, but the most common approach is to use ranging sensors mounted on the robot to form *occupancy grids* or equivalent [21, 23]. These divide the area of the environment into a regular grid, and associate a binary state (or potentially a floating probability) to each cell according to whether they are penetrable or not [8, 32]. Other approaches avoid this metric division of space and favour topological mapping [17, 32] but this is ill-suited to the methods presented herein.

Since navigation is the primary goal, the accurate shape of objects, or even their identification and classification, is not important. Similarly, a certain level of error is tolerable, since encountering an unexpected obstacle is no more serious than to cause delay to the journey of the robot.

In many senses, the aims of robotic exploration are to discover areas where objects do *not* exist and to plot pathways through them. In context-aware computing environments, however, the interest lies in the position and type of the objects themselves.

3. Ray-tracing in Static Worlds

The authors have previously developed and demonstrated a method for using rays within positioning systems to discover and characterise static objects [14], which will be briefly summarised here.

A *ray* is established between a mobile transmitter and a receiver when the ranging measurement (time-of-flight or equivalent) for a signal propagating between them is *not* rejected by the positioning algorithm. Rays describe straight pathways which are unobstructed to the positioning medium. Given a diverse series of transmitter locations over time, and a similarly diverse distribution of receivers, rays penetrate into the environment and obstructions become apparent from low densities of rays. This is the premise for using ray-tracing to create and maintain the world model.

Large numbers of rays are best stored within an *accumulation grid*. This segments the volume of the environment into a regular three-dimensional grid, each cell having an associated voting count (initially zero). A new ray is quantised onto the grid by incrementing the voting count of each and every cell it intercepts. After a period of time, the voting grid can be converted to a binary occupancy grid via thresholding. By introducing the voting stage it is possible to account for any noise in the system. More complex approaches using probabilistic beams are possible, but can involve greater processing requirements and are not necessary to demonstrate the underlying methodology (a brief discussion of the probabilistic approach is given in [14]).

Once an occupancy grid is established analysis is necessary to autonomously identify the presence of objects. Three-dimensional region growing can be used to extract containment volumes and shapes. It is also possible to use techniques specialised to the particular object. For example, the shape and height of horizontal surfaces has been extracted using *profile plots*. These plot the ratio of perimeter and area against vertical height, and are described in detail in [14].

This approach works well in mapping new regions with static objects. A static environment allows an accumulation of a large number of sightings without fear that the eldest results are no longer applicable. Real environments are more dynamic, however, with objects continually shifting. In the form described, accumulation grids offer no temporal dependence and cannot adapt easily to dynamic environments.

4. Dynamic Environments

There are two major tasks that a context-aware system must perform to maintain synchronisation of the world model with the real world:

1. Observing and reacting to the disappearance of objects.
2. Observing and reacting to the appearance of objects.

4.1. Disappearing Objects

Observation of the disappearance of modelled objects can be achieved using ray-tracing techniques. Removal of an object permits the passage of rays through the volume it previously occupied. A system must therefore search for the passage of rays through space that the world model asserts to be occupied. A system can model an object's spatial extents in a series of ways, the most important of which are listed below.

Point modelling. The object is modelled as a three-dimensional point in space.

Exact or boundary modelling. The exact three-dimensional shape of the object is known and stored as a set of vertices, faces, or edges.

Primitive or generic modelling A generic model for a class of objects is created. Each instance of the model can vary in scale and dimension.

Constructive solid geometry. Geometric primitives (cubes, spheres, etc.) are combine in a series of geometric translations and boolean operations to create a shape that closely approximates that of the object.

Spatial enumeration. Essentially a three-dimensional occupancy grid. Grid voxels are assigned a state corresponding to whether or not they contain (part of) the object.

Extent modelling The overall extents of the object are stored, ignoring intricacies in the specific contours. This is essentially a bounding box for the exact model.

Today's systems tend to employ primitive or extent modelling, or a combination thereof. This necessitates allowance for error when asserting that a ray/object intersection has occurred. In particular, intersections near the edge of an object model are rarely sufficient evidence to assert disappearance. Such intersections can easily result from inaccuracies in the model or in the calculated position of the tracked object.

To deal with error it is instructive to examine how close an intersecting ray passes to the centre point of the object model whilst remaining within its bounds. Here, 'centre' is defined as the centre-of-mass of an object of uniform density with the specific shape of the model. This is consistent with the intuitive idea of the centre. Such an approach makes the assumption that the object is sufficiently convex that its centre-of-mass lies within its volumetric bounds. If this is not the case, the object can be subdivided into smaller, convex sub-objects. Once the point of nearest approach is determined the distance between it and the centre point is calculated. This value is compared with the distance from the centre point to the model edge or face along a line extending from the centre point toward the point of nearest approach (Figure 1). When small, the ratio of these two distances implies a ray passing deep within the object, rather than glancing it. High values of this *distance ratio* have ambiguous interpretations due to the errors inherent in both the ray and the model.

4.2. Appearing Objects

The appearance of an object results in a *lack* of rays penetrating the associated volume. Dynamic environments require an extension to the occupancy grid approach described in Section 3, which will allow temporal evolution of the initial occupancy grid. This evolution can be based on one of two ideologies:

Temporal. New results are stored in a temporary accumulation grid. At regular time intervals this grid is analysed and the base occupancy grid updated if necessary.

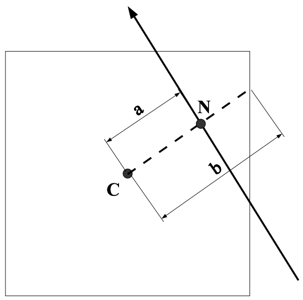


Figure 1. Intersection of a ray with a cubic object. The centre is at C, the point of nearest approach at N. The distance ratio is a:b

This gives a synchronous update of cell state across the coverage area.

Spatial. New results are again stored in a temporary voting grid. Updates to the base occupancy grid are performed at times derived from the new sightings in the area and only for that specific area. This produces an asynchronous update of cell state.

If a region receives no new data during an update period of the temporal approach, it is not possible to determine whether this is due to the appearance of an object or simply because no sightings were made in the vicinity. Hence the temporal approach is useful only in a system which guarantees near-uniform sighting densities between such updates. This is not applicable to a personnel tracking system since people do not move to uniformly cover the area they inhabit.

The spatial approach does not necessarily suffer from this problem. Its implementation uses two accumulation grids; one to collect the ray information as before, and one to collect the rays that are *expected* given the sightings and the system characteristics. We term these rays *pseudo-rays* and the corresponding grid the *pseudo-grid*.

When a cell within the pseudo-grid reaches a certain vote threshold, the *pseudo-threshold*, its occupancy state is re-evaluated based on the corresponding ray count in the voting grid. This decision can be based on the ratio of the number of intersecting rays and the number of intersecting pseudo-rays, the *intersection ratio*. This ratio should be less than or equal to 1.0 since each ray should correspond to exactly one pseudo-ray. For an occupied cell, a very low ratio (below 0.5) is expected whilst unoccupied cells should exhibit larger values.

Having allocated an occupancy state to a cell we compare the state to the corresponding base occupancy grid state and update this as appropriate. Following this, the corresponding cell is zeroed in both accumulation grids. Figure 2 illustrates the process in two dimensions. Here, cells with a pseudo-grid count of three or greater are updated, resulting in a total of five cells being compared.

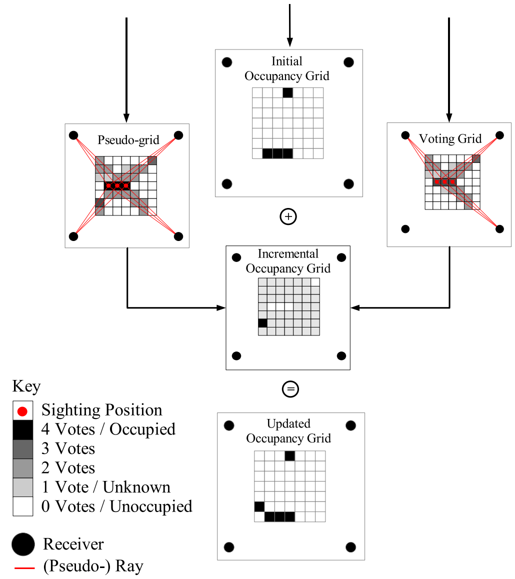


Figure 2. Evolving the occupancy grid

The spatial approach is superior to the temporal approach. Since it updates specific cells only when they are likely to have useful state information, it is more efficient and less prone to error.

4.2.1. Parameter Selection and Responsiveness The pseudo-threshold for the spatial approach must be chosen to balance the conflicting desires for high update rate, reliability, and coverage. A small value for this parameter creates a more responsive system but leaves it more susceptible to statistical anomalies which could be smoothed out with increased data. Practically, then, it is necessary to trade-off between response time and error in the world model. The ideal value for a given system will be strongly related to the update rate and error characteristics of that system, and is best determined on a per-system basis.

The response speed of a dynamic system implementing the spatial approach is dependent on three major factors. The first is the update rate of the positioning system itself. Faster update rates result in a faster accumulation of rays and thus a more responsive system in general.

A second factor is the sighting distribution. In a typical installation the system has little or no input as to where the transmitters go. Rather, they are attached to personnel who move with motives inconsistent with creating a uniform distribution of sightings. If no transmitter visits a particular region, no updates can be made. This is generally acceptable, however, since a lack of sightings implies that the region has not been visited, and is unlikely to have been reconfigured.

As previously mentioned, the final factor is the pseudo-threshold parameter which dictates when a cell is to be updated.

4.3. Dealing with Translations

Translation of an object can be seen as a combination of detecting the disappearance and appearance of two objects, followed by the recognition that the objects are the same.

Recognition can be achieved using the profile plot described in [14]. A Hidden Markov Model [27], or similar, can use the plot to classify and label the objects after a training period. Alternatively, complementary sensor systems may be available to identify objects.

5. Implementation and Results

The ideas presented above were implemented using the ultrasonic Bat system. This system uses multilateration techniques to determine the position of a mobile transmitter to within 0.03m. Ultrasonic signals propagate from transmitters to a series of static, ceiling-mounted receivers.

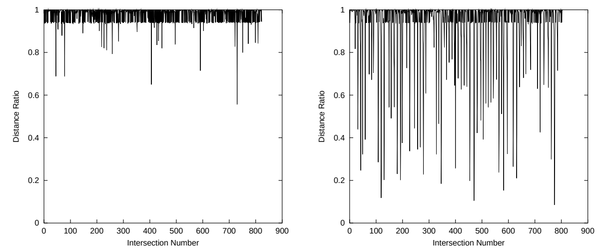
Successful ray-tracing requires that rays penetrate into all areas of interest. Ceiling-mounted positioning systems thus require that the mobile transmitter be positioned *below* any object, in order that rays might intersect it.

Bat transmitters are worn by personnel, usually at chest height. Thus rays propagate from chest height to ceiling and do not typically intercept objects. As such, the Bat system has been optimised as a personnel positioning system, and is not ideally suited to ray-tracing in its current incarnation. Nonetheless, it provides sufficient information to act as a testbed for the methods herein.

5.1. Tracing Rays for Disappearances

Computer monitors provided objects with which to examine the usage of rays when searching for disappearances. The SPIRIT system [3] models the position and rotation of computer monitors within the Laboratory for Communication Engineering (LCE). The modelling is purely primitive: no specifics are stored for individual monitors. Rather, each monitor was treated as a cube of dimensions 0.5m$\times$0.5m$\times$0.5m (in the case of the LCE, this was a good approximation since all monitors were similarly sized and styled).

As an initial test, a tracked user was allowed to work in front of a particular monitor for 3 minutes. The monitor was then removed, and the experiment repeated. In each



(a) Monitor correctly modelled

(b) Monitor removed

Figure 3. Distance ratio variation

case, the rays that intersected the model of the monitor were recorded. Figure 3 shows the distance ratio variation for the two situations.

With the monitor in place, a high density of intersections with a ratio greater than 0.9 was observed. The minimum observed ratio was 0.556, the average was 0.975, and the standard deviation of the ratio was 0.0426.

However, once the monitor was removed the ratio was seen to vary more dramatically, with a minimum of 0.0850, an average of 0.921, and a standard deviation of 0.170. The most revealing characteristic is the frequency with which the distance ratio was less than 0.8 (0.9% of readings with the monitor in place, 11.8% of readings with the monitor removed).

Based on this information, three fake monitors, labelled A, B, and C, were added to the world model. The aim was to autonomously highlight the fake monitors based on signals travelling through the Bat system. Each new positioning result was evaluated for an intersection with nearby monitors. If the distance ratio was below 0.5, the data for the following 100 intersections of the monitor were recorded and the proportion with distance ratio less than 0.5 was determined.

Results were collected over a period of two days. Including the three fake monitors, 49 monitors were modelled across the laboratory. Note that not all laboratory members were present during the test period and consequently no sightings were made in the vicinity of their associated desks.

In total, 32 monitors were flagged for monitoring. Eight had a non-zero proportion of distance ratios less than 0.5. The remaining 24 either saw no distance ratios less than 0.5, or never built up 100 intersections within the two days.

Monitor A was found to have a 54% proportion of distance ratios less than 0.5, and monitor B 52%. Insufficient data was available for monitor 3 due to a sparse sighting distribution in its vicinity. In addition, two further monitors, X and Y, were found to have proportions of 15% and 14% respectively. On investigation, X was found to have moved significantly, and Y had been removed altogether. The highest recorded proportion for a correctly modelled monitor was 5%.

These results are successes, despite the fact that only a primitive model was used to crudely model varieties of monitor.

5.2. Spatial Approach for Appearances

To use the spatial approach, the Bat system was characterised using a straightforward model. Ultrasonic signals were assumed to have a maximum propagation distance of 5.0m from the transmitter, derived empirically from archived positioning data. For each sighting, the receivers within this distance were determined and the pseudo-rays (propagating from the cell containing the sighting to the cells containing these receivers) quantised onto the pseudo-grid.

It was necessary to add a filtering level to the occupancy grid to produce more robust results given the relatively small amount of data. The final occupancy grid was filtered by removing all cells marked as occupied but with all neighbouring cells unoccupied. This effectively asserted that all objects of interest extend over at least two cells. This is a reasonable assertion for an office environment and a cell size of the order of 0.2m.

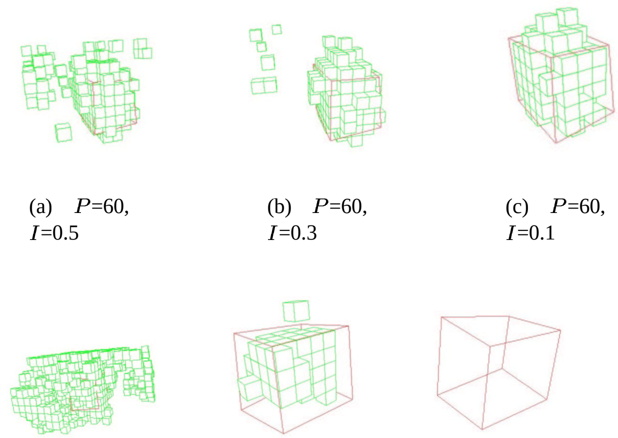
5.2.1. Dynamic Parameter Values A large cardboard box was suspended in air and used to evaluate the creation of an occupancy grid using the pseudo-grid method. A Bat transmitter was moved randomly by a human around the area of the box to build up the necessary sightings.

Figure 4 illustrates the occupied cells resulting from the collection of 5 153 sightings (50 843 rays) near the large cardboard box (shown in outline) for different values of the pseudo-threshold and the intersection ratio threshold.

Figures 4(a)-(c) demonstrate the need to correctly set the intersection ratio threshold. When too large (Figure 4(a)), the threshold results in premature assignment of the occupied state to some cells. A small ratio of the order of 0.1 will ensure a reliable state assignment (4(c)).

The pseudo-threshold is similarly important. When too small, cell states are evaluated very regularly but evaluations are necessarily made on a low volume of data, resulting in large errors (Figure (4(d)). Care must also be taken to ensure that the threshold is not excessively large. The result shown in Figure 4(f) is due to very few cells reaching the threshold using the data from the 5 153 sightings. Thus, all cells lie in an undetermined state.

Figure 5 shows the evolution of the occupied cells as more rays are analysed. The precise evolution is dependent



(a) P=60, I=0.5 (b) P=60, I=0.3 (c) P=60, I=0.1

(d) P=3, I=0.1 (e) P=100, I=0.1 (f) P=1000, I=0.1

Figure 4. Effect of pseudo-threshold (P) and intersection ratio threshold (I)

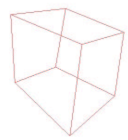
on many factors, including the sighting distribution, the object size, and the cell size.

5.2.2. Variation of Intersection Ratio The Bat system was used to evaluate the dynamic ray-tracing methods by attaching a Bat transmitter to a small autonomous vehicle. The vehicle was programmed to move forward until an impact was registered on its front bump sensors and then reverse a small distance and rotate by a random amount, before repeating the process. This algorithm results in an unpredictable distribution of sightings that is not optimised for ray-tracing (and therefore approximating real personnel positions).

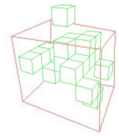
Two large tables, A and B, were positioned at random within a room (Figure 6(a)). The autonomous vehicle was allowed to collect results freely for approximately 30 minutes, creating data set I. Table B was then shifted to a new position (Figure 6(b)), and the experiment repeated to create data set II.

Four cells at the table height of 0.7m were selected at random such that two were occupied by a table surface (Cells (7,18) and (7,19)) and the other two unoccupied (cells (5,10) and (14,18)). Figure 7 shows the variation of the intersection ratio with each update of the relevant cell, using data set I. Figure 7(b) shows solely the occupied cell variations for clarity.

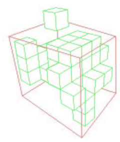
The occupied cell (7,18) initially exhibits an intersection ratio of 0.0 where the pseudo-vote is non-zero but the ray vote is zero. After approximately 850 pseudo-votes, a ray



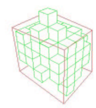
(a) 25,000 rays



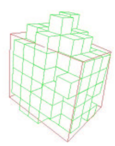
(b) 30,000 rays



(c) 35,000 rays

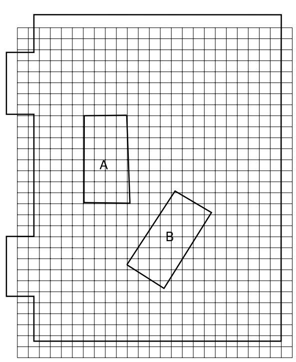


(d) 40,000 rays

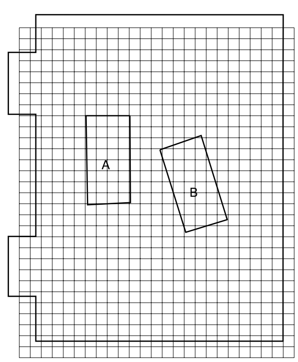


(e) 50,000 rays

Figure 5. Evolution of the occupancy grid for a pseudo-threshold of 60 and an intersection ratio threshold of 0.1

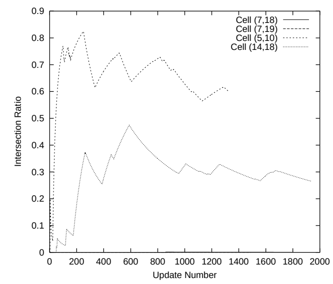


(a) Initial setup



(b) Final setup

Figure 6. Room, table, and grid setups (2D projections)



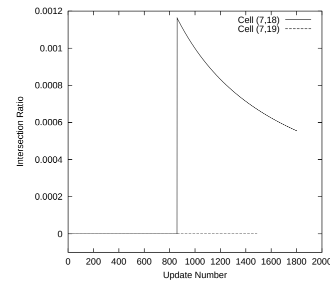


Figure 7. Variation of intersection ratio with cell update

is erroneously established that passes through this cell. This is attributable to noise in the system. Thereafter, updates do not involve votes for rays, only pseudo-rays, and the intersection ratio follows an inverse linear law. This pattern is characteristic of an occupied cell. The cell (7,19) did not erroneously establish a ray and maintained a ratio of 0.0 throughout.

The unoccupied cells do not show a definite pattern. Here, a much higher proportion of the pseudo-rays should be accompanied by rays, resulting in a highly variable intersection ratio. The ratio variation appears essentially random, and has a significantly larger value than the corresponding occupied ratio. This is the basis of the spatial method.

These Figures assume an infinite pseudo-threshold. As described above, we use the threshold to segment the grids over time and update the corresponding occupancy grid accordingly. Figure 8 shows the variation of the intersection ratio with a pseudo-threshold of 200 for both an occupied cell (cell (7,18)) and an unoccupied cell (cell (5,10)). The dynamic approach re-evaluates the cell occupancy state at every multiple of 200. The correct states in this case can be derived using an intersection ratio no greater than 0.2.

5.2.3. World Model Creation Data set I was used to construct an occupancy grid from an initially empty state. A pseudo-threshold of 200 and a intersection ratio threshold of 0.05 were used. Figure 9 illustrates the occupancy of cells at the table height (0.7m).

Even with the relatively small amount of data, the method can correctly derive features given no initial data. The signal to noise ratio can be increased with a larger volume of data.

5.2.4. Dynamics Demonstration of the dynamic method in a realistic scenario was possible using both data sets I and II, respectively collected before and after table B was moved. The aim was to observe an adaptation of an established occupancy grid as new data was input.

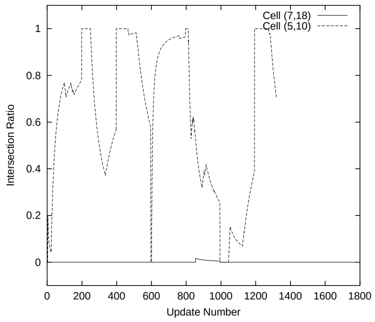


Figure 8. Intersection ratio variation with a pseudo-threshold of 200

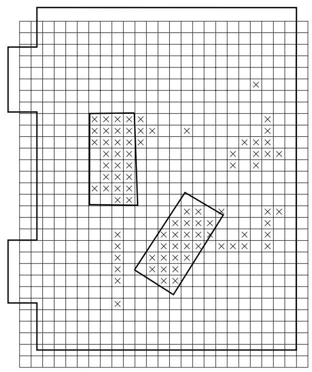


Figure 9. An experimental occupancy grid. Crossed cells are occupied

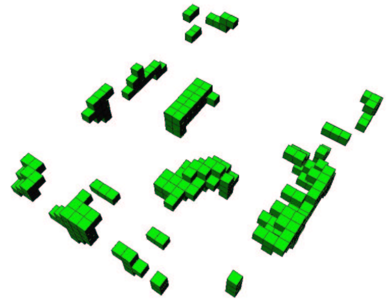


Figure 10. An experimental occupancy grid (three-dimensional)

An occupancy grid was initially established using the entirety of the data set I. This created an accurate occupancy grid to approximate the grid expected to exist after a large number of small updates. Figure 10 illustrates the occupancy grid after this setup phase. The two tables of Figure 5 are visible in the centre of the grid, whilst further occupied cells are shown around the room outskirts. These cells are due both to genuine objects stacked against the walls, and the known limitations of ultrasonic ray-tracing (see [14]). The area of primary interest is central within the grid.

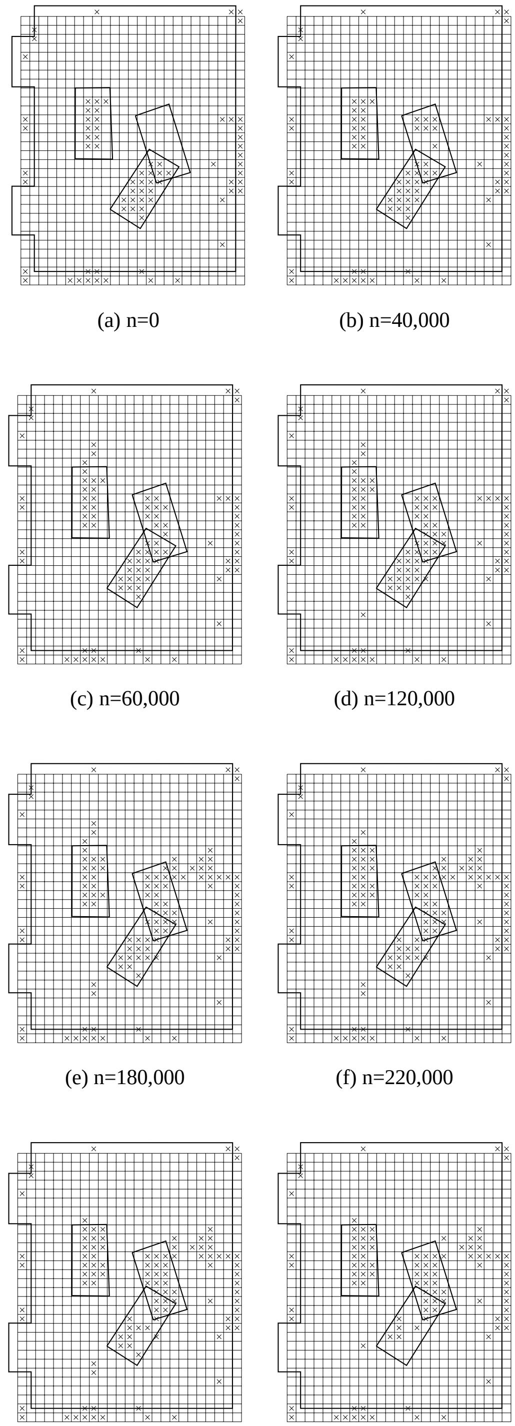
Data set II was added using the dynamic method with a pseudo-threshold of 200 votes, an intersection ratio threshold of 0.05, and a cell size of 0.2m. Figure 11 shows the resultant evolution of the occupancy grid with sightings at the table height. A gradual shift of occupied cell density, consistent with the movement of table B, is apparent.

6. From Grids to Models

Whilst an occupancy grid is a sufficient world model for autonomous navigation purposes it is generally insufficient for context-aware applications. Such applications must have a higher level understanding of the grid, which can be difficult to supply autonomously.

The trade-off made between update rate and accuracy means that the occupancy grids produced by a useful dynamic system typically contain too much noise for robust object recognition based on shape or volume. Instead, the method is very useful for verifying the current model and highlighting inconsistencies. It is envisaged that users of the environment will continually provide spatially diverse positioning data with which to perform a ray-tracing analysis and determine local (in)consistencies between the real world and the world model. Application of the methods described allows for specific areas of inconsistency to be pinpointed. For example, the appearance of a group of cells in a room believed to be empty of objects provides constraints that can be used by other sensor systems. A robot could be directed to investigate the area when the room is empty, video cameras could be directed at the area for vision analysis, or RFID readers could be energised in the vicinity to search for new RFID tags. Most important is the potential to spot disappearances. An object in the real world that does not have a representation in the world model leads to a failure to derive context; an object in the model that is no longer in the real world leads to an incorrect context which can have worse consequences.

As an example, consider a context-aware system which automatically unlocks and starts the engine of a car when the owner approaches. If the car is unknown to the model, no context is inferred and the car does nothing as the user approaches. Now consider the car having moved to another car park on the other side of the building, and the world model not having been updated. When the owner walks past the first car park, her car unlocks and the engine starts in the other car park. The first scenario (failure to spot appearance)



(a) n=0 (b) n=40,000

(c) n=60,000 (d) n=120,000

(e) n=180,000 (f) n=220,000

(g) n=260,000 (h) n=280,000

Figure 11. Occupancy evolution for number of rays, n. Both table positions are shown

is frustrating for the owner, whilst the second (failure to spot disappearance) has the potential to be far more than frustrating. When other sensor systems are not available to perform a detailed spatial analysis of a region, the inconsistency information can be used to reduce trust in the region. The system may choose not to unlock and start a car if some minutes earlier another user apparently walked through the outskirts of the space believed to be occupied by the car.

7. Conclusions and Further Work

This paper has presented an extension to the idea of tracing ray pathways through an environment in order to spot the appearance, disappearance, and movement of objects. The method uses existing infrastructure in an environment augmented with an accurate positioning system. Experimental results using the Bat system have shown that the method is viable and can respond to changes in dynamic environments. Such a response is necessary for a context-aware computing platform, since an accurate world model is required for accurate context determination using position. Once an accurate occupancy grid is formed using the method, it can be integrated with further sensor inputs to identify or classify objects and update the live world model.

The methods presented here are relevant for any positioning system based on the propagation of signals that can be obstructed by physical objects. If such systems are to be ceiling-based, it is necessary to design the system such that direct signals propagate into the areas of interest. This can be achieved by ensuring the transmitter remains below the height of the objects of interest, or by moving some receivers to lower heights, away from the ceiling. It is hoped that future work will examine ways to integrate sensor information into a robust world model that can cope with the appearance, disappearance and reconfiguration of objects.

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] A. Bystrom et al. Advanced wavelan positioning system. SMD116 Program Project, May 2001.

[2] 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.

[3] 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.

[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] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM 2*, pages 775–774, 2000.

[6] P. Bahl, V. N. Padmanabhan, and A. Balachandran. Enhancements to the RADAR user location and tracking system. Technical report, Microsoft Research, February 2000.

[7] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven A. Shafer. Easyliving: Technologies for intelligent environments. In *HUC*, pages 12–29, 2000.

[8] 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.

[9] A. Dey. *Provididng Architectural Support for Building Context-Aware Applications*. PhD thesis, Georgi Institute of Technology, 2000.

[10] E. Foxlin, M. Harrington, and G. Pfeifer. Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Application. In *Proceedings of {SIGGRAPH} 98*. Addison-Wesley, 1999.

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

[12] 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.

[13] R.K. Harle and A. Hopper. Using Personnel Movements For Indoor Autonomous Environment Discovery. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 125–132, March 2003.

[14] Harle R.K and Hopper A. Building World Models By Raytracing Within Ceiling-Mounted Positioning Systems. In *Proceedings of UbiComp, Seattle, Washington, US*, October 2003.

[15] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. *Mobile Computing and Networking*, 1999.

[16] J. Hightower and G. Borriello. Location sensing techniques. *IEEE Computer*, August 2001.

[17] 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.

[18] 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.

[19] J. Y. Lee and R. A. Scholtz. Ranging in a Dense Multipath Environment Using an UWB Radio Link. *IEEE Journal on Selected Areas in Communications*, 20(9), December 2002.

[20] M. Weiser. The Computer for the 21st Century. *Scientific American*, 1 September 1991.

[21] M.C. Martin and H.P. Moravec. Robot Evidence Grids. Technical Report CMU-RI-TR-96-06, Carnegie Mellon University, 1996.

[22] S. Mitchell, H. Naguib, G. Coulouris, and T Kindberg. A QoS Support Framework for Dynamically Reconfigurable Multimedia Applications. In *Second IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems, Helsinki, Finland*, June 1999.

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

[24] 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.

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

[26] 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.

[27] L. R. Rabiner. A tutorial on hidden Markov Models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, February 1989.

[28] 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.

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

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

[31] A. Smailagic and D. Kogan. Location sensing and Privacy In A Context-Aware Computing Environment. *IEEE Wireless Communications*, 9(5):10–17, October 2002.

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

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

[34] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci. The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 1–10, 1999.

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

[36] M.A. Youssef, A. Agrawala, and A.U. Shankar. WLAN Location Determination via Clustering and Probabuility Distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pages 143–152, March 2003.