Acquiring Medium Models for Sensing Performance Estimation

Aman Kansal, James Carwana, William J Kaiser, and Mani B Srivastava
Department of Electrical Engineering
University of California, Los Angeles
Email: {kansal,jcarwana,kaiser,mbs}@ee.ucla.edu

***Abstract*— The quality of sensing in practical sensor network deployments suffers due to the presence of obstacles in the sensing medium. If such unknown obstacles are present, and the sensor data indicates that no targets of interest are detected, then there is no easy way for the application to distinguish between the cases that there really is no target or that the targets are located in occluded regions. The obstacles may not be known before deployment and may change over time. Hence, it is of interest to develop methods which enable a sensor network to determine the presence and extent of sensing occlusions. We present one such method based on the use of a range sensor to map the obstacles in the medium. A network architecture to support efficient medium mapping facilities is presented, along with several design choices in the acquisition and update of the medium map data. We also present algorithms to rapidly acquire this data and share it among multiple nodes. All algorithms presented are implemented on prototype hardware consisting of an actuated laser and an embedded processing platform.**

I. Introduction

Since their conception, sensor networks [1] are finding applications in a variety of problem domains spanning security, automation, scientific exploration, education, and entertainment. One key network performance metric that affects all these applications is the uncertainty in the data collected by the underlying transducers. The factors that lead to this uncertainty may be divided into two classes:

- **Intrinsic Factors.** The signal acquired by the transducer is only a noisy version of what exists in the environment and below a certain signal to noise ratio, the uncertainty induced due to noise may render the signal useless. Next, this acquired signal may degrade further within the system. First, an information loss typically occurs during quantization in the analog-to-digital converter. Second, depending on network conditions, part of the data may be lost due to communication channel errors or network congestion. Third, in certain applications, such as those using image sensors, the collected data may have to undergo lossy compression in order to meet the bandwidth constraints, resulting in a further degradation.
- **Extrinsic Factors.** The signal of interest generated in the environment must travel through some medium to reach the sensor transducer. The medium may attenuate it and the extent of attenuation tolerated depends on the transducer noise. Also, obstacles in the medium may block the signal entirely. These obstacles create occluded regions which are not sensed by any of the sensors in the network. While the resource cost and performance tradeoffs for the intrinsic factors may largely be under the control of the system designer, the deployment environment and presence of obstacles is usually not.

In this paper we address the extrinsic factors. Our goal is to provide methods for sensor networks to actively measure the sensing medium and determine the presence of obstacles. Acquiring such a medium model has many benefits.

First, the model may be used to determine the occlusions to sensing and provide an estimate of sensing uncertainty. One such method to model the sensing capacity of a sensor network given the occlusions was provided in [2].

Second, the network may attempt to overcome the occlusions. For instance, if some of the nodes are mobile, they may be moved in order to minimize occlusions. Even with static nodes, in certain applications it may be possible to use the medium model to determine choke points which a target must cross if it has to enter the occluded regions. Data from such choke points may be used to estimate presence of targets in occluded regions. Further, energy resources being spent on sensing at nodes whose transducers are occluded may be saved,

and devoted to communications, aggregation or other network activity.

Third, the information on occlusions for each sensor may help guide the aggregation process when data from multiple sensors is combined. For instance, a simple strategy may be to ignore data from nodes which indicate a detection in regions known to be occluded from their view.

Additionally, the medium maps may also act as secondary data inputs for higher layer applications to help determine context. For instance, the presence of several tree-trunk shaped obstacles may indicate to the application that the sensor network is operating in an outdoor region while the presence of wall and ceiling like obstacles may help it infer that the network is operating indoors. Such context information can be used in several ways.

Sensing performance has traditionally been characterized in terms of a sensing range and transducer noise. We believe that explicit medium information will help sensor networks provide better performance guarantees than those based on circular disk models alone.

Environment mapping is commonly studied in robotics for robot navigation and path control. The use of context information has been explored in various other domains such as pervasive computing and location aware services. However, the nature of context information and mapping constraints differ in the case of sensor networks. We will note the commonalities and differences in subsequent sections.

A. Paper Organization

The next section presents our proposed system architecture, designed to integrate medium mapping resources into the sensor network. It also lays out the key design issues which arise in this design, some of which are addressed in subsequent sections. In particular, section III discusses strategies to reduce the time for mapping by exploiting the structure of the environment and section IV considers issues in efficient representation, exchange and update of medium maps in a distributed manner. Section V summarizes related work and section VI concludes.

II. System Architecture

A. Enabling Self-Awareness

We propose a sensor network architecture in which supplementary sensors are added for medium mapping. These sensors may not be required for the sensor network application directly, and we refer to them as *self-awareness sensors*. The other sensors are referred to as application sensors. Figure 1 depicts a block diagram of the proposed architecture. Much of this architecture design is motivated by the system concerns that arose in the development of our prototype system, and thus, the design is tightly coupled to practical considerations of its use cases.

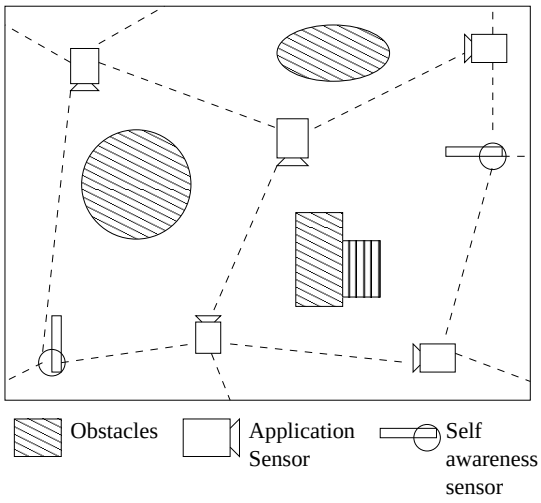


Fig. 1. System architecture, consisting of self-awareness and application sensors.

The figure reveals one of our key system design choices: that the self-awareness sensors need not be present at the same nodes as the application sensors. There are several reasons for this choice. Firstly, the self-awareness sensors may be required to be deployed at a vastly different density than the application sensors. In our system for instance, the application sensor has a range of 7.3m [1], while the self-awareness sensor has a range of more than 100m. For a given set of obstacle locations in the medium, the numbers of the two types of sensors required may well be different and thus, adding a self-awareness sensor at every node might be an unnecessary overhead. Secondly, the medium dynamics may be much slower than the dynamics of the phenomenon sensed by the application sensors. In this case, a few mobile self-awareness sensors could acquire and update the medium map [3] while many more application sensors actively track the rapid dynamics of phenomenon of interest.

[1]Our application sensor is a camera and this range is calculated in an occlusion-free environment based on a minimum required spatial resolution of 1cm per pixel.

It may be noted that supplementary sensors have been used in other ways to reduce sensing uncertainty – such as the use of magnetic sensors to help augment the vehicle tracking performance of a video sensor network. However, the use of self-awareness sensors is not quite the same thing as the use of multiple sensing modalities. In multi-modal sensing, the data fusion process is highly application dependent. If the primary sensor data, the images in the above example, are used for another application which is not interested in magnetically active objects, that supplementary sensor is not of relevance. In our approach, the application processing is independent of the self-awareness sensors; the medium information is used to estimate or improve the quality of application sensor data, and any application which uses the sensor data benefits from this.

There are several design issues in the acquisition and sharing of self-awareness data. Before discussing these in detail, we describe the prototype system that we developed for providing a concrete context to explore this problem space.

B. System Hardware

The self-awareness sensor node in our prototype test-bed consists of:

1) a laser range sensor, Leica Disto Pro [4],
2) a pan-tilt platform, Directed Perception PTU-D46-17N, [5], to move the direction of the laser beam, and
3) an embedded processing platform, Intel Xscale based Stargate [6].

The laser ranger is interfaced to the Stargate using a serial port, over which the control commands are sent to acquire range readings from it. The pan-tilt platform is also interfaced using a serial port, for sending it commands to change its orientation. Since the Stargate has only one directly accessible serial port, we added a PCMCIA to serial port card to the Stargate for creating another serial port. The prototype is shown in Fig. 2.

The application sensors used are network cameras with pan, tilt and zoom capabilities. Thus, these cameras may reconfigure their orientation using the medium information provided by self-awareness nodes. The re-configuration process itself is an interesting research problem, but is not addressed in this paper.

The remaining components in the test-bed are passive objects of various shapes and sizes which limit the sensing range of the application sensors. These are the obstacles in the medium to be mapped by the self-awareness sensors. Since we have only fabricated one self-awareness node as yet, we emulate multiple self-awareness nodes by placing the single node at multiple locations.

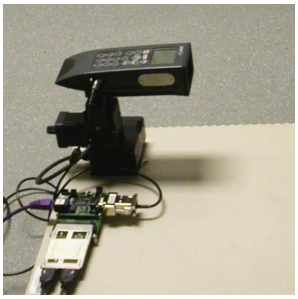


Fig. 2. Prototype self-awareness node.

C. System Design Issues

Conceptually, the medium mapping process can be viewed as this: the self-awareness sensors collect data about the medium characteristics and then provide this data to the application sensors. Now, there arise several issues in the process of collecting and distributing such self-awareness data with minimum resource overhead.

First, the range data must be collected as fast as possible, in order to enable faster updates and to minimize the node active time. The naïve method would be to take a range measurement at a sufficiently fine granularity and determine which space elements in view are blocked. However, the environment has certain structure to it and it is intuitive to expect that if such structure can be exploited, we can reduce the number of readings to be collected, thereby speeding up the data acquisition process. Implementing this procedure on a practical laser ranger and pan-tilt platform reveals the various issues in doing this. Our aim is to determine useful strategies to minimize data acquisition time, which are expected to work in a wide variety of environments in view of the system limitations.

Second, the medium map has a significant data size. This motivates a study of efficient methods for sharing this data among the application and self-awareness nodes. The storage and communication strategies may attempt to minimize the data size, but the compression techniques should not lead to extensive processing overheads for retrieval and update. The storage strategies for a data set typically depend upon the nature of queries that must be answered using the stored data. Our objective is to determine the relevant strategies to store and share medium map data in a manner that enables answering

queries about the medium occlusions for the application sensors.

These issues are addressed in the next two sections. There are further design steps required to complete the self-awareness process. The medium information may have to appropriately converted to a form that enables the application sensor to determine which portions of its field of view are occluded. For instance, in our test-bed the spatial coordinates of an obstacle may have to be mapped to the corresponding pixel coordinate in the image taken by a camera. For the case of cameras, this process is known as external calibration and existing methods [7] can be used for this step. Also, self-awareness data from multiple nodes has to be combined to develop a coherent medium map. In our test-bed we assume that each node is localized, in a common coordinate system. Such an assumption is valid for sensor networks since most nodes are static and localization is a basic service required for many other sensor network applications. Given the location information, the medium data from different locations can be combined using the appropriate spatial transformations and we do not dwell on it further.

D. Technology Choice

We have selected a laser range sensor to measure the medium characteristics. There are other means to achieve the same functionality. One alternative is to use stereo-vision to derive depth maps. This approach has limitations of accuracy compared to a laser sensor and may miss small obstacles in the environment. Also, stereo-vision algorithms require detecting key features in the environment which can then be identified in the multiple images collected by the cameras in the stereo-system. It may not work in all environments where such key features are hard to detect. Further, the stereo-vision generated depth map is relative and a known dimensional measure is required in the environment to scale the depth measurements correctly. These issues motivated us to use a range sensor. Among range sensors, there are various options, such as ultrasound ranging and infra-red ranging. Both these alternatives give lower accuracy than the laser, and the distance range of the infra-red sensors is usually lower than that of laser based ones. However, both these are usually available at a lower cost, and may be of interest in scenarios where the low accuracy is not a deterrent. The map acquisition algorithms discussed in later sections are in fact applicable to these types of range sensors as well. The data sharing methods developed in our prototype are applicable to all the above alternatives.

Mapping based on range sensors has also been considered in mobile robot navigation. While our problem is similar, the solution is different and the resultant design issues are distinct. For a robot, medium mapping is a primary task necessary for navigation. Even if the environment map was available a-priori, the range sensor on a robot is actively used to avoid obstacles in the path. In the case of a sensor network, the demands from the self-awareness sensor are less stringent. First, the self-awareness data is not being used for active navigation of a mobile robot. Each node need not have a self-awareness sensor. The static nature of most sensor nodes also enables carrying out the mapping process at a much slower pace than required for a robot. A direct consequence of this is in the technology choice – the laser range sensor used in our work is significantly cheaper than those typically used on robots and has a slower response time for taking range readings. The pan-tilt actuation used in our prototype is much slower than required for real time navigation, and can be achieved with lower energy expenditure. Second, the assumption on the availability of localization also causes our work to differ significantly from robotic mapping. In the case of a team of multiple robots exploring an unknown environment, the processes of localization and mapping are dependent [8] on each other. Combining the maps generated at multiple robots is not straightforward and different techniques such as maximum sub-graph isomorphism have been explored for that purpose. On the other hand, for sensor networks, if localization is available as is needed for many sensor network applications, combining the self-awareness data from multiple nodes is simpler. Thus, the medium mapping problem in sensor networks is more tractable than the corresponding robotics problem, and we expect to be able to solve it with a lower resource overhead.

III. Adaptive Medium Mapping

There are three basic types of maps [9]:

1) Grid based: These maps divide the space into discrete elements which are mapped as empty or occupied [10], [11].
2) Feature based: These maps are a collection of landmark features in the environment [12], [13].
3) Topology based: These maps represent distinctive places in the environment as nodes in a graph and the interconnections or paths between them as the edges [14], [15].

For the purpose of determining sensing occlusions, the most relevant maps are the grid based ones. These

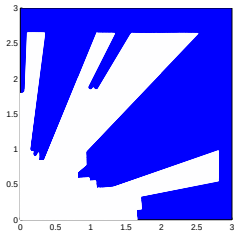
provide a detailed model of the sensing medium, based on which the application sensors may estimate their sensing performance.

The process to generate a grid based medium map proceeds as follows. Range measurements are taken by orienting the laser at varying pan and tilt from a fixed location to form a depth map for the space around the self-awareness node. Such depth maps from multiple self-awareness nodes may then be combined to develop a medium map for the region of deployment. Figure 3-(a) shows a small section of a possible deployment environment with obstacles of various shapes. Figure 3-(b) shows one horizontal plane in the depth map generated by scanning the laser placed at one corner of the region, and Figure 3-(c) shows a similar scan from a node placed near another corner. The white regions in the scans represent clear field of view while the dark regions represent occluded areas. Note that the second node does not scan the entire quadrant scanned by the first one. Both nodes scan 90 degrees. Combining the two scans for this small section, in the plane represented, yields the medium map shown in Figure 3-(d). The occluded regions could either be obstacles or spaces between obstacles which are empty but not accessible from the laser locations.

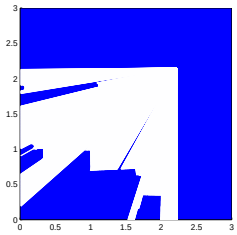
Suppose the medium map is stored as a three dimensional matrix where each entry represents a voxel (a small volume element) in space. Each entry in the matrix denotes whether the corresponding voxel is empty, occupied or unknown. Such a matrix is referred to as an occupancy grid (OG). The data collection task is to take range readings at different pan and tilt angles and determine which voxels are empty or occupied. Voxels not accessible from a self-awareness sensor node's position remain unknown in the OG generated at that node, and have the same effect as occupied voxels in the calculation of clear field of view. Entries in the OG are filled using the range readings taken by the laser range sensor at different pan and tilt angles. At each orientation, the voxels along the free line of sight of the laser are designated as empty.

Let us now consider the practical considerations which affect the time taken for collecting the laser range data. Our experience with the laser device used in our prototype indicates that it must be stationary when acquiring a range reading, for accuracy of measurement. If the laser beam is continuously moved, the device still returns range measurements. However, the time taken to acquire a range reading increases by an unpredictable amount, which makes it is impossible to determine the

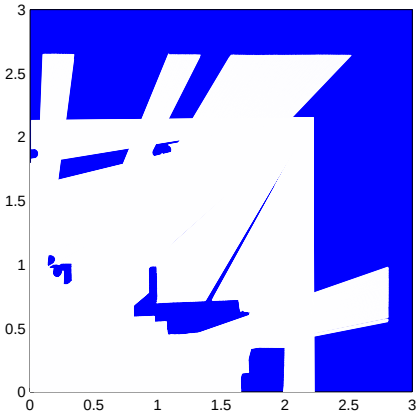
(a) Sample Deployment Environment



(b) Depth map A



(c) Depth map B



(d) Medium map

Fig. 3. Map generation process.

orientation, and hence the relevant voxels in the OG, which corresponds to the returned range reading. Thus, the laser is moved in small steps along the pan and tilt axis and the readings are acquired when the device is stationary. With this, the time taken, T, to acquire a reading at a particular pan and tilt position can be modeled as consisting of three components:

$$T = t_{move} + t_{axdx} + t_{read} \quad (1)$$

where t_{move} represents the time taken by the pan-tilt platform to move to the next position at a constant speed using the pan and tilt motors, t_{axdx} represents the additional time taken due to acceleration and deceleration phases in executing the motion step since constant

speed cannot maintained throughout the motion, and t_{read} represents the time taken for the laser range sensor to acquire a reading.

The parameters t_{move} and t_{axdx} depend on the capabilities of the pan-tilt platform, and t_{read} depends on the laser unit, and the reflectivity and the angle of incidence of the surface at which the laser beam is pointed. For a constant pan and tilt range scanned by the laser at a given granularity, suppose the total angular distance moved is d and the number of readings taken is N. The total time taken to execute the scan becomes:

$$T_{scan} = d * t_{move} + N\{t_{axdx} + t_{read}\} \quad (2)$$

The time taken thus varies with N. If we reduce N, and thus increase the angular step size at which the laser stops to take a reading, T_{scan} can be reduced. Our objective is to generate the OG in a manner that exploits the structure in the environment in order to reduce the number of readings taken by the laser.

A. Reducing the Scan Time

For many signals, a frequency spectrum analysis yields that the signal needs to be sampled only at a course granularity. Considering the range scan in space as the signal to be sampled (such as shown in Fig 3-(b,c)), indicates that the Nyquist sampling interval is quite small due to the presence of sharp discontinuities in the environmental features, and the value of N derived in this manner is very high. Thus, we need to exploit other forms of structure in this signal. We propose two heuristics to exploit such structure. These are described in terms of the pan motion of the laser[2] and the goal is to minimize the number of readings for a constant pan scan range.

1) Adaptive Filter Based Algorithm: The first method is based on the use of an adaptive filter to learn the medium structure from the readings acquired and when the laser readings match those predicted based on the adaptive filter, the scan step size may be increased to reduce the number of points measured. We refer to this method as adaptive filter based (AFB) algorithm and use least-mean-square (LMS) adaptation as our adaptive filter. LMS is a gradient descent based heuristic for the filter adaptation. Suppose the filter at time step i is represented by a length n column vector $\mathbf{h(i)}$, the previous n values of the scan are represented by a column vector $\mathbf{x}(i)$, and $e(i)$ is the error in the prediction, calculated using:

$$e(i) = \mathbf{h^T}(i)\mathbf{x}(i) - x(i+1) \quad (3)$$

[2]Similar methods apply to tilt motion.

where $x(i+1)$ is the measurement at time step $(i+1)$. Then the gradient search equation to minimize the square-error becomes:

$$\mathbf{h}(i+1) = \mathbf{h}(i) - \mu \bigtriangledown (|e(i)|^2) \quad (4)$$

where the gradient $\bigtriangledown(\cdot)$ is taken with respect to filter coefficients $\mathbf{h}$, and μ is a constant step size for the gradient descent. Simplifying the above expression leads to a computationally efficient filter update equation:

$$\mathbf{h}(i+1) = \mathbf{h}(i) - \mu e(i)\mathbf{x}(i) \quad (5)$$

We deviate slightly from the conventional LMS filter in that the step size at which readings are taken varies as we adapt the angular step size using the AFB algorithm. Whenever the prediction error $e(i)$ is below a threshold, we increase the angular step size for the laser pan motion, as described in the next subsection. The step size cannot be increased beyond a certain limit which depends on the size of the smallest feature of interest to be mapped. The problem of adaptive filter design is not addressed as part of this work; several adaptive filters are available, and may be explored depending on the specific needs of the system [16].

2) Adaptive Linear Prediction Algorithm: The second method is more domain specific and exploits the nature of the signal being sampled. Since adaptive filters are not specially designed for depth maps, they require significant learning data. However, observe that when sampling at a fine granularity, many of the surfaces in the medium, especially for indoor built environments, can be approximated as consisting of small planes, or in one plane of pan motion, a curve consisting of small line segments. When scanning this curve, the number of samples taken need only be sufficient to reproduce the line segments. Thus, when range data indicates that the laser is taking measurements along a line, we may increase the angular step size of the pan motion.

This prediction method proceeds as follows: Determine the coordinates of the points on the sampled curve corresponding to the previous two range readings. Use these to determine the equation of the line joining them. Now predict the coordinates of the next point along this line, which the laser will hit when moved by its current angular step size. At the next laser pan step, calculate the error between the predicted point and the measured point. If the error is below a threshold, increase the angular step size. The exact algorithm to control the step size appears in the next subsection. We will refer to this method as the adaptive linear prediction (ALP) algorithm.

Some further improvements can be applied to both the AFB and ALP algorithms as described below.

First, when the laser range extends beyond the area of interest, such as when the laser is near an edge of the deployment area, we need not scan the exact structure of the medium and may increase the angular step size to the maximum limit dictated by the minimum obstacle size to be detected. The step size would be reduced as soon as a feature of interest is detected within the region of interest.

Second, for a given angular step size, the scan resolution in space varies with distance from the laser. This can be seen in Fig. 4: the spatial resolution $\Delta\alpha$ is higher than $\Delta\beta$ for the same angular step $\Delta\theta$. Thus, $\Delta\theta$ may be increased when scanning obstacle A, as the OG is stored at a constant resolution in space.

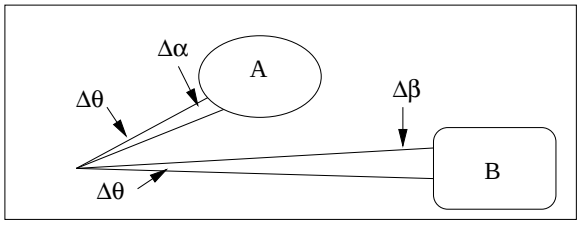


Fig. 4. Varying spatial resolution with obstacle distance.

Third, when the step size has been increased using an adaptive method and a high error is encountered, it is likely that a large change in the scanned depth occurred in the region between the current and the previous readings. Thus, if we back-track, i.e., move the laser back to the previous position and then scan the region of large change using a small step size, the quality of reproduction can be improved significantly with a small increase in the number of readings taken.

B. Algorithm Implementation and Evaluation

The precise ALP and AFB algorithms derived from the above discussion are stated together in Fig. 5. The key difference among the two algorithms is in the prediction step. The values of parameters used are summarized in Table I. The parameter $\Delta\theta_0$ represents the minimum angular step size required to achieve a desired spatial resolution, for the maximum laser range in the environment. The step size μ was chosen to be 0.03. Larger step sizes were tried, and while they yield faster convergence and somewhat lower error, they lead to divergence of the LMS filter in some cases. The filter length was kept at $n = 3$. A larger filter length can learn more complex shapes but needs a longer adaptation

1) **Initialization:** Set $\Delta\theta = \Delta\theta_0$. Take first M range measurements by panning the laser in steps of $\Delta\theta$. Set $k = 0$, $i = 0$.
   If method is AFB, initialize adaptive filter **h** to zeros. M = length(**h**).
   If method is ALP, $M = 2$.
2) **Prediction:** Predict the next point, based on previous M readings (using AFB or ALP methods described in previous subsection).
3) **Measurement:** Pan laser by $\Delta\theta$ and take next reading.
4) **Update:** Calculate error $e(i)$ between the predicted point and measured point.
   IF $e(i) < e_{thresh}$:
   a) Set $k = k + 1$.
   b) Set $\Delta\theta = \Delta\theta_0 + k\delta$
   c) **IF** $\Delta\theta > \Delta\theta_{max}$, Set $\Delta\theta = \Delta\theta_{max}$
   d) For AFB, update **h** using equation (5).
   ELSE
   a) Set $k = 0$.
   b) Set $\Delta\theta = \Delta\theta_0$.
   c) **Back-track:** Move laser to previous pan position.
5) If scan not completed, set $i = i + 1$, and go to step 2.

Fig. 5. Adaptive step size control algorithm for medium mapping.

time and the signal characteristics may change over such time as the laser moves from one obstacle to another with different a shape. Optimizing the LMS filter for various environment categories is part of future work. We now evaluate these algorithms by comparing them

Parameter	Value
e_{thresh}	1 cm
μ	0.03
$length(\mathbf{h})$	3
δ	0.05°
$\Delta\theta_{max}$	$\Delta\theta_0 + 1^\circ$

TABLE I
ALGORITH PARAMETER VALUES.

to a constant step size approach, both using simulations, which allows testing over several obstacle placements, and using experiments on our prototype hardware.

1) Simulations: To test the proposed methods, we generate several random obstacle placements in a square region, place the laser ranger in one corner of the

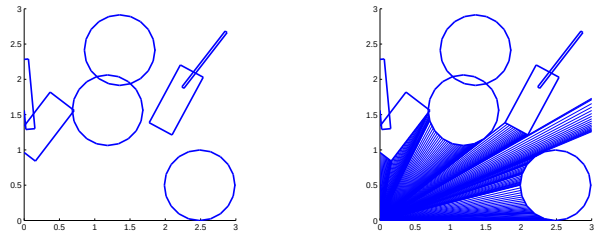
square and simulate the range readings that would be collected when the laser is panned with the step size computed as per the adaptive methods. As a base case, the readings are also generated if the laser were to be panned with a periodic step size kept at $\Delta\theta_0$. The obstacles generated are rectangles and circles, placed randomly and overlapping with each other leading to arbitrary shaped obstacles, such as shown in Fig. 6(a). Ten such random obstacle topologies were generated. A simulated laser scan using adaptive step sizes is shown in Fig. 6(b), where the varying step size between successive pan positions is apparent, and the back-tracking near obstacle edges may be seen as well. The angular step size increases as the filter learns a curve and falls back again when the curve changes significantly.

Error performance of a scan is calculated as follows. From the application perspective, the OG generated from the range scan data is of interest. A high resolution periodic scan is first carried out and the OG generated from it is used as the ground truth. Then, for each simulated scan, the corresponding OG is generated and the error is measured in terms of the number of voxels which differ with respect to the ground truth OG. A two dimensional OG is used for the simulations, considering only the pan motion of the laser. The exact same techniques can be applied in the tilt direction as well to generate the three dimensional OG. The number of differing pixels is normalized by the number of total pixels in the OG, and the error is plotted as a percentage.

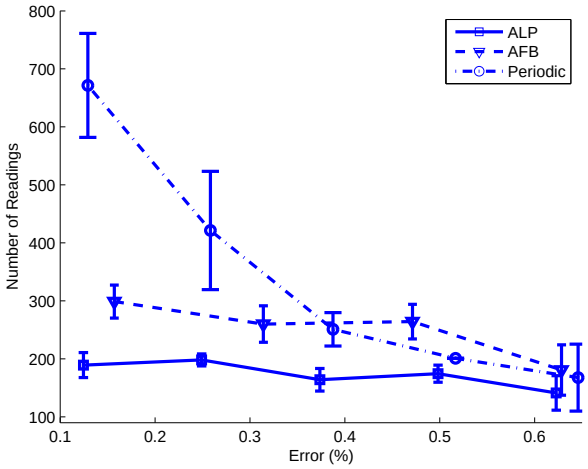
For each topology, multiple scans are carried out with $\Delta\theta_0$ varying from $0.09°$ to $0.9°$. Increasing $\Delta\theta_0$ implies reducing the resolution of the scan and thus increasing the error. For each scan, we count the number of readings taken when a periodic scan is used and when the adaptive algorithms are used. To compare the savings in time when using the adaptive methods, we plot the number of readings taken for the same error using the periodic and adaptive schemes.

The simulation data needs to be averaged over the random topologies. Since each topology may lead to different error values with varying $\Delta\theta_0$, the number of readings, N, required cannot be directly averaged. We divide the error axis into small intervals and consider the average of number of readings taken for error across multiple topologies divided into these intervals. These averaged plots are shown in Fig. 6(c). The error-bars show the standard deviation across multiple topologies.

The figure shows that at low error the adaptive scans require much fewer readings than the periodic scan. Among AFB and ALP, ALP performs better since it



(a) Sample deployment (b) Sample AFB scan



(c) Error performance

Fig. 6. Comparing the number of readings taken with varying error performance for the three scanning strategies.

is designed specifically for the nature of the sampled signal. As the angle step is increased for the periodic scan, the error increases and the number of readings reduces. However, the advantage due to adaptive methods dwindles as the step size for the periodic scan has also increased. The portions of the scan where the adaptive methods can use an increased step size now form a smaller fraction of the total scan.

2) Experiment: We now experimentally compare the performance of these algorithms using our prototype self-awareness node. Apart from validating the implementation of our proposed methods on an embedded platform, the experiments include additional effects not modeled in the simulations. First is the presence of sensing errors in the laser readings and pan motion. Secondly, while in the simulation, only the number of readings could be plotted, in the experiment we can measure the total savings in time. These includes the time for motion and the time for taking a laser reading. The time for taking a laser reading varies with the color of the surface scanned and even for the same number of

readings, the difference in the positions of those readings on some dark colored spots in the environment may cause an increase in time consumed; such effects were not modeled in the simulation.

The environment used in our experiment was a $2m$ square space with some objects placed as obstacles. The results are shown in Fig. 7. The better adaptive method, ALP, was only used.

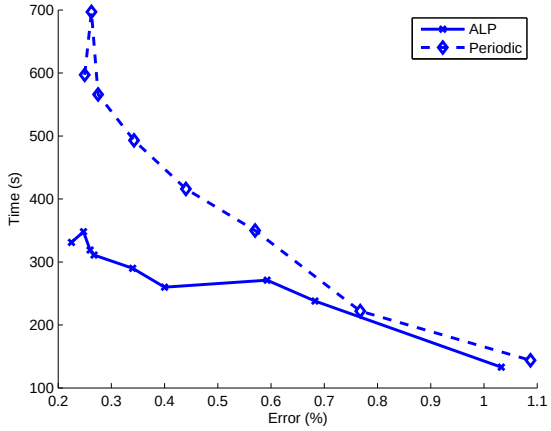


Fig. 7. Comparing the adaptive and periodic scan strategies using the prototype self-awareness node.

These results are similar to those seen in simulations. The evaluation clearly shows that significant savings in time can be achieved using the adaptive methods.

Apart from the savings in time, another side effect of the adaptive methods is a saving in storage and communication overhead as discussed in the next section.

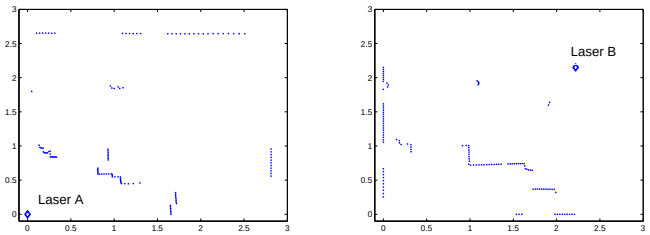
IV. Sharing the Self-awareness Data

After each laser sensor has collected the range scan, using either a periodic scan or the adaptive methods, the data at each self-awareness node must be used to determine the medium occlusions at each application node. There are several design choices in doing this, differing in how the data is stored and how the depth maps from multiple laser sensors are combined.

Data-sharing schemes depend on the nature of queries that need to be answered based on the stored data. In our system, the application sensors are only interested in determining the occluded regions from their point of view. The individual occupancy grid (OG) generated by a laser scan (such as seen in Fig. 3(b)) may be stored at each laser node. To generate the medium map all laser nodes with overlapping fields of view must transmit their OG's to a common location where the medium map is then computed. The individual OG's are first translated along the spatial axes in a global reference frame, depending on the laser coordinates and then an 'OR' of their overlapping voxels is computed. The OR operation assumes that the occluded voxels (obstacle surfaces or unknown) are stored as zeros and the clear regions as ones. This generates a combined OG, referred to as the medium map (as was seen in Fig. 3(d)). Relevant portions of such a medium map once generated can be distributed to each application sensor.

A. Computing the Field of View at Application Sensors

We can improve the above process if we observe that the medium map need not be calculated centrally for the application sensors to be able to calculate their occlusions. Each self-awareness node need not compute its OG, but may just store the raw coordinates of the points scanned by the laser ranger. Let us refer to these points as range coordinates. The range coordinates which were used to compute the OG's seen in Figures 3(b) and 3(c) are shown in Figures 8(a) and 8(b). We



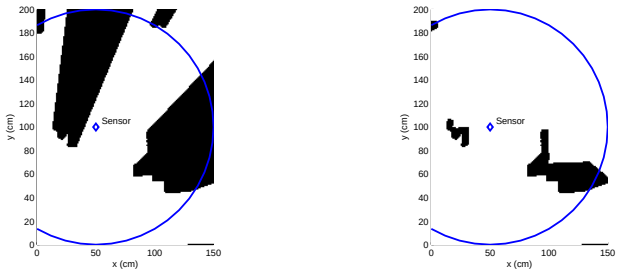
(a) Range coordinates A (b) Range coordinates B

Fig. 8. Range coordinates used to compute occupancy grids (OG's)

present a distributed method using which the raw range coordinates at the individual laser nodes are directly used by the application sensors to determine their occlusions.

Suppose the nominal range of the laser is known to be R_l, and that of the application sensor is R_s; the actual ranges may be much shorter due to presence of obstacles. An application sensor now access those laser nodes which are within R_l distance, and queries the range coordinates stored by the laser, as well as the coordinates of the laser itself. The application sensors now computes its own OG's, to represent which voxels in their field of view are unoccupied, as follows: connect the laser node coordinates to each of the range coordinates, and all voxels through which the connecting lines pass are deemed to be unoccupied. The application sensor only generates the OG for a circular area of radius R_s around it and not the entire R_l.

As an illustration, consider the $3m \times 3m$ area seen in Fig. 3 and the corresponding range coordinates in Fig. 8. Laser A was placed at $(x, y) = (0, 0)$ and laser B was placed at $(2.22, 2.15)$. Suppose an application sensor, say, a camera with full 360° pan range, is placed at $(0.5, 1)$, and has $R_s = 1$. It first accesses the range coordinates from laser A and computes its local OG by checking which pixels are unoccupied, as shown in Fig. 9(a). The OG is computed at a resolution of $1cm$, which means that each grid point in the OG is separated by $1cm$. Next, it accesses the range coordinates



(a) OG, using laser A (b) OG, using lasers A and B

Fig. 9. Example: Generating a combined OG directly at the application sensor.

from laser B. Again, it connects the range coordinates with the laser coordinates and computes the unoccupied pixels, thus discovering further pixels in its OG which are unoccupied. This OG is seen in Fig. 9(b).

B. Evaluation

In our proposed approach, no global medium map is generated. This approach has several advantages compared to the laser nodes first generating their OG's, exchanging them, generating the medium map and then distributing it to the application sensors.

Memory Requirements: The application sensors calculate the OG only within their sensing range, R_s. The size of the OG is proportional to the cube of R_s (or the square of R_s when only the horizontal plane is considered). If $R_s < R_l$, which is the case in our system, this means that each application sensor has to compute a much smaller OG than at a laser node. This is significant, since embedded nodes have limited memory, and increasing the memory size in increases their power consumption and cost.

Communication and Storage Overhead: The communication overhead in our approach is only the data-size of the range coordinates. This is significantly lower than the OG at a laser, and the combined OG's (medium maps) generated using the data from multiple lasers. For instance, at a resolution of $1cm$, the size of the OG matrix in two dimensions is 10^4 binary values per square meter, while the range coordinates only store a few points corresponding to obstacles in the same area. The number of points stored can further be reduced using the adaptive methods discussed in the previous sections.

It may be noted that the OG need not be communicated in its raw form but could be compressed using a data compression or image compression method. Our approach of storing range coordinates can be viewed as a domain specific compression scheme which is expected to perform better than the general compression techniques. Fig. 10 shows the storage space required for our proposed approach with Periodic, AFB, and ALP methods, as well as that required for storing the OG when using a data compression technique: TIF-packbits compression, which is a lossless compression method exploiting the repetition of bits, and an image compression technique: JPEG, which uses a transform to derive an efficient data representation. The sizes are for the same data sets as generated in the previous section.

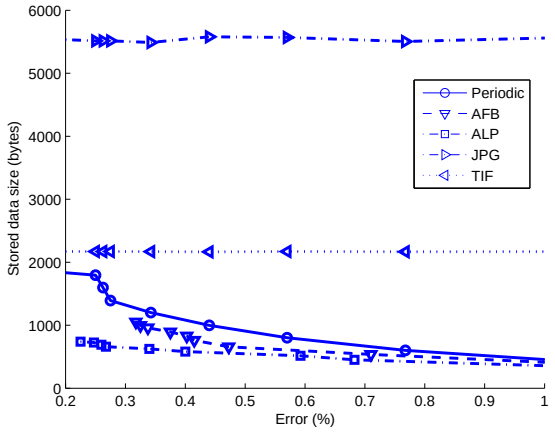


Fig. 10. Comparison of data size for multiple storage strategies, with varying scan error.

Our storage approach clearly has lower data size. Also, the general compression methods operating on the OG as a whole are unable to achieve any significant reduction in size as the tolerable reproduction error is increased by reducing the number of range coordinates. Reduced data size saves communication energy, bandwidth, and the energy required to write to flash memory.

A trade-off in this approach is that for an application sensor, the time to compute the OG from the range coordinates is larger than the time to decompress a

medium map image if sufficient memory is available to hold the entire medium map. However, the computation of the OG from range coordinates is carried out over only the application sensor's range, a much smaller region than the medium map generated by the laser nodes. Thus, computation time is not significant, and when memory is limited, it may in fact be lower than the image decompression time. Further, computation costs are typically lower in sensor nodes than the costs of flash memory writes and communications, and hence this approach is still preferred.

Resolution Flexibility: The application sensor is free to compute its field of view at a much lower resolution than the data generated at the laser. Without application information, the laser node may have to compute the medium map at the highest resolution possible with the collected data, and generate a much larger OG matrix than required.

Reduced Remote State Dependence: The proposed method is completely distributed, and dependence on the state of remote nodes is minimal. Firstly, no data exchange is required among the self-awareness nodes. Since R_l may be large, such communication may have required data to be sent over large distances. Reducing this overhead saves significant bandwidth. Secondly, the application sensors are not required to know the state of progress of the self-awareness nodes coordination and determine if the combined medium map has been generated. They may access each self-awareness node's data asynchronously. Such reduced state dependence not only simplifies design in distributed systems but also helps improve system robustness because in case of a node's failure, its effect on other nodes is reduced. The modular and decoupled operation yields a significant advantage when the sensor network is executing multiple applications and services.

Efficient Updates: The medium may change over time and our data sharing method allows for efficient updates. First, transmitting only the range coordinates to the application sensors reduces the size of data communicated. This approach has at least as much advantage in communication overhead as seen in Figure 10 with respect to the size of the update data to be exchanged. Second, the asynchronous data access mode, rather than having to wait for the self-awareness nodes to generate a coherent map, allows for update information to be incrementally incorporated with minimal delay as and when the medium changes.

The updates may be exchanged using either a push or a pull approach. In the push approach, the laser nodes send an update to the application sensor whenever a change is detected and in the pull approach, the application sensor queries the laser nodes for updates. Depending upon the nature of the deployment, one of these methods would be preferred. For instance, if the medium changes infrequently, then the self-awareness nodes may push the updates. This saves the overhead of the application nodes periodically polling for updates when not many updates are expected. However, if medium changes are frequent, the push approach will lead to a high overhead. In this case, the application sensors should query for updates at the frequency they require. The pull approach is also useful for power management as it allows application sensors to enter low power modes without worrying about medium changes when they are not even sensing. In our system we have selected the pull approach to keep the update process at each application sensor decoupled from the self-awareness node's operation. The laser periodically collects data and publishes it at a known location. In our prototype, we use the common HTTP protocol for data exchange. The range coordinate data is stored in a web accessible folder on the self-awareness node. The application sensors access this data as required.

There are other enhancements possible to the update process if the overhead of maintaining remote nodes' state is acceptable. For instance, each self-awareness node may maintain the locations of the application sensors which access it. Then, it could compute which portions of its range coordinate data affect each application sensor. In case of updates to the range coordinates data, the laser would compute which application sensors are affected and publish a flag indicating such change. Now, when an application sensor accesses a self-awareness node, it first looks at the update flag corresponding to itself. If the flag indicates that the data has changed, it downloads the new data, else the unnecessary communication cost of the redundant update is eliminated.

V. Related Work

While the use of medium characterization methods specifically for sensor networks is little explored, several related aspects of the problem have been researched. The most closely related problem is that of mapping in robotics, such as discussed in [11], [10], [8] among many other works. Our problem differs as we do not need to support simultaneous mapping and localization. Also, since localization may be decoupled from mapping in sensor networks, the problems of combining the medium maps from different sensor nodes differs significantly

from robotics. Further, the sensor nodes are only interested in computing the occluded regions, rather than generating a global map for navigation. This allows us to use lower resource overheads and use simpler methods than required for robots.

Learning world models for enhancing computation performance has also been considered for context awareness and location aware computing [17], [18]. These are orthogonal to the medium characterization problem for sensor networks since the nature of queries about the world are very different. Those world models are designed to help the system determine where the user is, what activity the user is engaged in and the state of the objects in the user's environment. In sensor networks on the other hand, the query of interest is the location of occlusions for the sensor. The use of medium mapping was also considered in [3]. However, adaptive methods to optimize the scanning process were not considered. Also, their system involved only one mobile mapping node and distributed mapping and data sharing methods were not relevant.

Another related body of work is the design of adaptive filters for predicting a signal from its previous values [16]. We use these methods in our specific context of adapting the sampling step size. We also propose a different prediction method specific to the problem domain using linear extrapolation. The use of varying sampling step size was also considered in [19] to reduce the number of samples taken but their techniques were not designed for mapping obstacles.

VI. Conclusions

We have considered the problem of determining sensing occlusions for allowing sensor networks to estimate the uncertainty in their coverage and when possible take recuperative measures. Measuring these extrinsic factors is a significant addition to developing noise models and other intrinsic system uncertainties, in order to control the sensing performance of sensor networks. We proposed distributed medium mapping methods and improved data collection and sharing procedures designed specifically for the nature of constraints in sensor networks. These methods were evaluated both in simulations and in experiments conducted on a prototype self-awareness node that we have built.

While this work forms an important first step toward realizing the autonomous operation of sensor networks with performance control, several new issues were revealed during our development process, which are yet to be explored in greater depth. One of these is the intelligent separation of mobile events from obstacles, so that depth maps are not generated based on transient occlusions such as due to human motion within the network. Further, the application specific trade-offs in the push vs pull methods for updates may be studied in more detail.

References

[1] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[2] Yaron Rachlin, Rohit Negi, and Pradeep Khosla. Sensing capacity for target detection. In *IEEE Information Theory workshop*, San Antonio, USA, October 2004.

[3] Robert K Harle and Andy Hopper. Building world models by ray tracing within ceiling mounted positioning systems. In *UbiComp 2003*, pages 1–17, Seattle, WA, USA, October 2003.

[4] Leica laser distancemeter. http://www.leica-geosystems.com.

[5] Directed perception. http://www.dperception.com.

[6] Stargate xscale processor platform. http://www.xbow.com/.

[7] Janne Heikkila and Olli Silven. A four step camera callibration procedure with implicit image correction. In *IEEE Computer Vision and Pattern Recognition*, pages 1106–1112, 1997.

[8] MWM Gamini Dissanayake, P Newman, S Clark, H F Durrant-Whyte, and M Csorba. A solution to the simultaneous localization and mapping (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.

[9] Howie Choset. The wide world of mapping. WTEC Workshop: Review of US Research in Robotics.

[10] H Moravec and A Elfes. High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation*, pages 116–121, St. Louis, MO, 1985.

[11] S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, 1996.

[12] R Smith, M Self, and P Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 1:167–193, 1990.

[13] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–298, 1992.

[14] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Twelfth National Conference on Artificial Intelligence*, pages 979–984, Menlo Park, July 1994. AAAI Press/MIT Press.

[15] D Radhakrishnan and I Nourbakhsh. Topological localization by training a vision-based transition detector. In *Proceedings of IROS 1999*, volume 1, pages 468 – 473, October 1999.

[16] S Haykin. *Adaptive Filter Theory*. Prentice Hall, 4th edition, 2001.

[17] G Chen and D Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000.

[18] G Biegel and V Cahill. A framework for developing mobile, context-aware applications. In *IEEE Percom*, Orlando, FL, March 2004.

[19] M Batalin, M Rahimi, Y Yu, D Liu, A Kansal, G Sukhatme, WJ Kaiser, M Hansen, GJ Pottie, and D Estrin. Call and response: Experiments in sampling the environment. In *ACM SenSys*, November 2004.