

Certainty Grids for Mobile Robots

H.P. Moravec
Carnegie-Mellon University
Pittsburgh, PA 15213

68
188114

CH 18052

1 Abstract

A numerical representation of uncertain and incomplete sensor knowledge we call Certainty Grids has been used successfully in several of our past mobile robot control programs, and has proven itself to be a powerful and efficient unifying solution for sensor fusion, motion planning, landmark identification, and many other central problems. We propose to build a software framework running on processors onboard our new Uranus mobile robot that will maintain a probabilistic, geometric map of the robot's surroundings as it moves. The "certainty grid" representation will allow this map to be incrementally updated in a uniform way from various sources including sonar, stereo vision, proximity and contact sensors. The approach can correctly model the fuzziness of each reading, while at the same time combining multiple measurements to produce sharper map features, and it can deal correctly with uncertainties in the robot's motion. The map will be used by planning programs to choose clear paths, identify locations by correlating maps, identify well known and insufficiently sensed terrain, and perhaps identify objects by shape. The certainty grid representation can be extended in the time dimension and used to detect and track moving objects. Even the simplest versions of the idea will allow us fairly straightforwardly to program the robot for tasks that have hitherto been out of reach. We look forward to a program that can explore a region and return to its starting place, using map "snapshots" from its outboard journey to find its way back, even in the presence of disturbances of its motion and occasional changes in the terrain.

2 Introduction

Robot motion planning systems have used many space and object representations. Obstacles have been modelled by polygons and polyhedral, or bounded by curved surfaces. Free space has been partitioned into Voronoi regions or, more heuristically, free corridors. Traditionally the models have been hard edged - positional uncertainty, if considered at all, was used in just a few special places in the algorithms, expressed as a gaussian spread. Partly this is the result of analytical difficulty in manipulating interacting uncertainties, especially if the distributions are not gaussian. Incomplete error modelling reduces positional accuracy. More seriously, it can produce entirely faulty conclusions: a false determination of an edge in a certain location, for instance, may derail an entire train of inference about the location or existence of an object. Because

they neglect uncertainties and alternative interpretations, such programs are brittle. When they jump to the right conclusions, they do well, but a small error early in the algorithm can be amplified to produce a ridiculous action. Most artificial intelligence based robot controllers have suffered from this weakness.

We've built our share of brittle controllers. Occasionally, however, we stumble across numerical (as opposed to analytic) representations that seem to escape this fate. One is deep inside the program that drove the Stanford Cart in 1979 [6]. Each of 36 pairings of nine images from a sliding camera produced a stereo depth measurement of a given feature, identified by a correlator, in the nine images. Some pairings were from short baselines, and had large distance uncertainty, others were from widely separated viewpoints, with small spread. The probability distributions from the 36 readings were combined numerically in a 1000 cell array, each cell representing a small range interval (Figure 1). Correlator matching errors often produced a multi-peaked resultant distribution, but the largest peak almost always gave the correct range. The procedure was the most error tolerant step in the Cart navigator, but it alone did not protect the whole program from brittleness.

A descendant of the Cart program by Thorne and Matthies contained a path planner [5] that modelled floor space as a grid of cells containing numbers representing the suitability of each region to be on a path. Regions near obstacles had low suitability while empty space was high. A relaxation algorithm found locally optimum paths (Figure 2). The program represented uncertainty in the location, or even existence, of obstacles by having the suitability numbers for them vary according to extended, overlapping, probability distributions. The method dealt very reliably and completely with uncertainty, but also suffered from being embedded in an otherwise brittle program.

Our most thorough use of a numerical model of position uncertainty is a sonar mapper, map matcher and path planner developed initially for navigating the Denning Sentry [4, 2, 3]. Space is represented as a grid of cells, each mapping an area 30 cm in some versions (50 centimeters on a side and containing two numbers, one the estimated probability that the area is empty, the other that it is occupied). Cells whose state of occupancy is completely unknown have both probabilities zero, and inconsistent data is indicated if both numbers are high. Many of the algorithms work with the difference of the numbers. Each wide angle sonar reading adds a thirty degree swath of emptiness, and a thirty degree

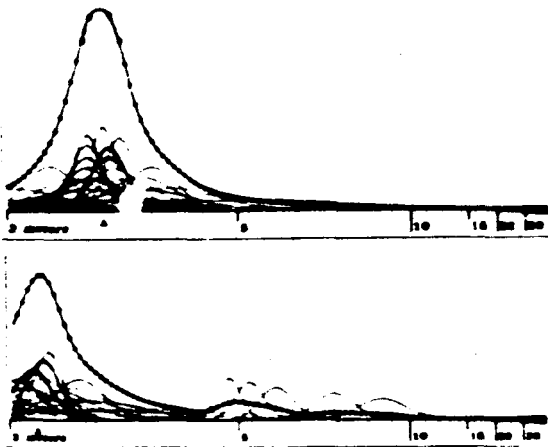


Figure 1: Nine Eyed Stereo - Identifications of a point on an object seen in nine different images taken as a camera traversed a track at right angles to its direction of view. Each pairing of images gives a stereo baseline, some short some long. Long baselines have less uncertainty in the calculated distance. The distributions for all 36 possible pairings are added in a one dimensional "certainty grid", and the peak of the resultant sum taken as the actual distance to the object. The top graph is for a case where all nine identifications of the point in the images are correct. The bottom is a case where one image is in error. The error produces eight small peaks at incorrect locations, but these are no match for the accumulation of the correct values.

arc of occupancy, by itself a very fuzzy image of the world. Several hundred readings together produce an image with a resolution often better than 15 centimeters, despite many aberrations in individual readings (Figure 3). The resiliency of the method has been demonstrated in successful multi-hour long runs of Denning robots around and around long trajectories, using three second map building and three second map matching pauses at key intersections to repeatedly correct their position. These runs work well in clutter, and survive disturbances such as people milling around the running robot.

Ken Stuart of MIT and Woods Hole has implemented a three dimensional version of the sonar mapper for use with small submersible craft. Tested so far only in simulation, but in the presence of large simulated errors, Stuart's program provides extremely good reconstructions, in a $128 \times 128 \times 64$ array, of large scale terrain, working with about 60,000 readings from a sonar transducer with a seven degree beam. Running on a Sun computer, his program can process sonar data fast enough to keep up with the approximately one second pulse rate of the transducers on the two candidate submersibles at Woods Hole.

Recently Serey and Matthies demonstrated the utility of the grid representation in a stereo vision based navigator [1]. Edges crossing a particular scanline in the two stereo images are matched by a dynamic programming method, to produce a range profile. The wedge shaped space from the camera to the range profile is marked empty, cells along the profile itself are marked occupied. The resulting map is then used to plan obstacle avoiding paths as with the stereo and sonar programs mentioned above (Figure 4).

Despite its effectiveness, in each instance we adopted the grid representation of space reluctantly. This may reflect habits from a recent time when analytic approaches were more feasible and seemed more elegant because computer memories were too small to easily handle numerical arrays of a few thousand to a million cells. I think the reluctance is no longer appropriate. The straightforwardness, generality, and uniformity of the grid representation has

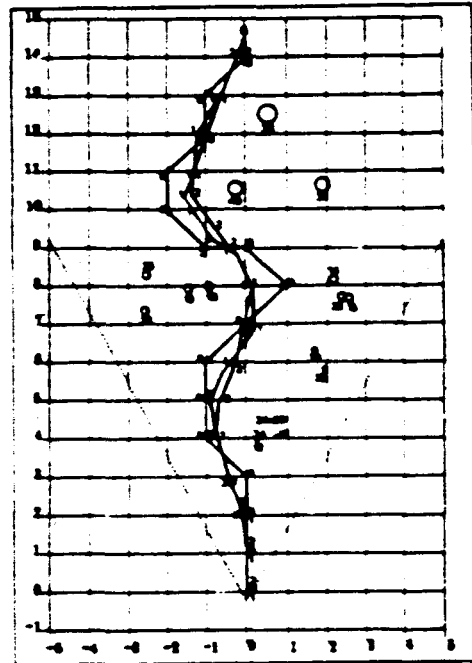


Figure 2: Relaxation Path Planner - A path is chosen that minimizes a given cost function in a Certainty Grid. Small perturbations are made in the vertices of the path in directions that reduce the cost.

proven itself in finite element approaches to problems in physics, in raster based approaches to computer graphics, and has the same promise in robotic spatial representations. At first glance a grid's finite resolution seems inherently to limit positioning accuracy. This impression is false. Cameras, sonar transducers, laser scanners and other long range sensors have intrinsic uncertainties and resolution limits that can be matched by grids no larger than a few hundred cells on a side, giving a few thousand cells in two dimensions, or a few million in three dimensions. Since the accuracy of most transducers drops with range, even greater economy is possible by using a hierarchy of scales, covering the near field at high resolution, and successively larger ranges with increasingly coarser grids. Besides this, the implicit accuracy of a certainty grid can be better than the size of its cell. The grid can be thought of as a discrete sampling of a continuous function. Extended features such as lines (perhaps representing walls) may be located to high precision by examining the parameters of surfaces of best fit. The Denning robot navigator mentioned above convolves two maps to find the displacement and rotation between them. In the final stages of the matching correlation values are obtained for a number of positions and angles in the vicinity of the best match. A quadratic least squares polynomial is fitted to the correlation values, and its peak is located analytically. Controlled tests of the procedure usually give positions accurate to better than one quarter of a cell width.

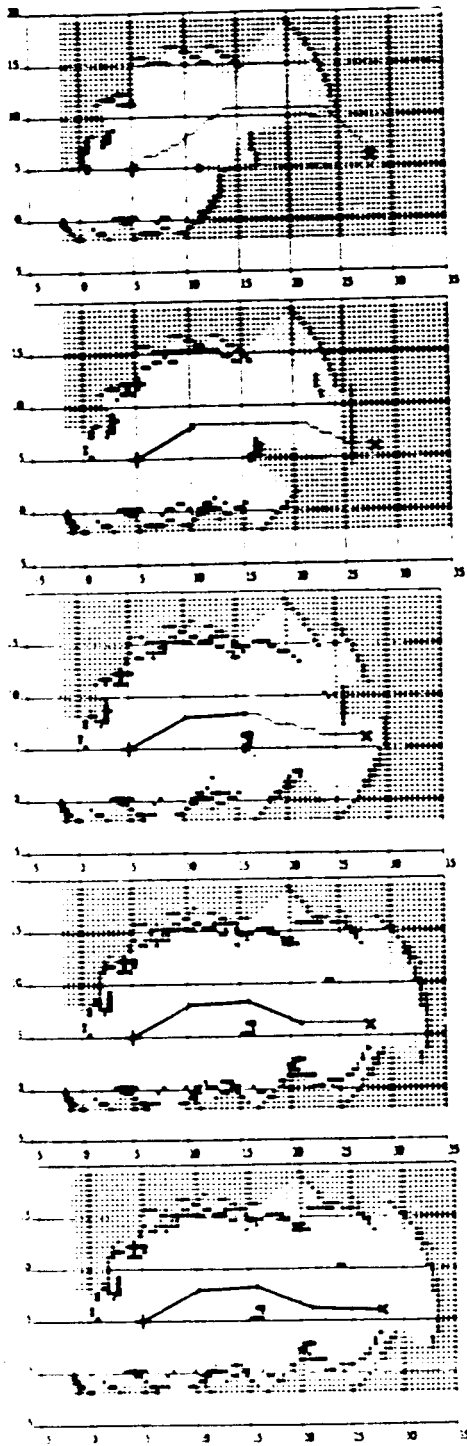


Figure 3: Sonar Mapping and Navigation - Plan view of the certainty grid built by a sonar guided robot traversing our laboratory. The scale marks are in feet. Each point on the dark trajectory is a stop that allowed the onboard sonar ring to collect twenty four new readings. The grid cells are white if the occupancy probability is low, dots if unknown, and \times if high. The forward paths were planned by a relaxation path planner working in the grid as it was incrementally generated.

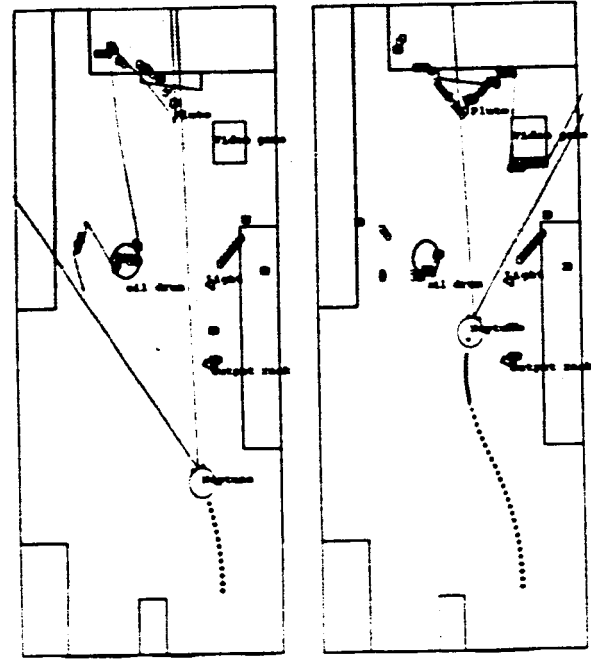


Figure 4: Stereo Mapping and Navigation - Plan view of the certainty grid built by a stereo guided robot traversing our laboratory. The situation is analogous to the sonar case of Figure 3, but the range profiles were gathered from a scanline stereo method using two TV cameras rather than a sonar ring.

Our results to date suggest that many mobile robot tasks can be solved with this unified, sensor independent, approach to space modelling. The key ingredients are a robot centered, multi resolution, map of the robot's surroundings, procedures for efficiently inserting data from sonar, stereo vision, proximity and other sensors into the map, other procedures for updating the map to reflect the uncertainties introduced by imprecise robot motion, and yet others to extract conclusions from the maps. We've already demonstrated procedures that produce local and global navigational fixes and obstacle avoiding paths from such maps. Other tasks, such as tracking corridors, finding vantage points with good views of unseen regions, and identification of larger features such as doors and desks by general shape seem within reach.

3 The Representation

The sonar mappers mentioned above are our most thorough use to date of the certainty grid idea. Although our original implementations used two grids to represent occupancy knowledge (labelled $P_{occupied}$ and P_{empty}), Stuart's 3D system uses only one. An analysis of the steps in our code reveals that one grid will indeed suffice, and this simplification makes clear several puzzling issues in the original formulation.

Before any measurements are made, the grid is initialized to a background occupancy certainty value, C_b . This number represents the average occupancy certainty we expect in a mature map, and encodes a (very) little bit of a-priori information we have about the world. In our lab a good C_b seems to be about the number of cells in the perimeter of the grid divided by the total cells ($4 \times 32, (32 \times 32) = 1.8$) in the case of the Denning code. If the space is very cluttered, C_b should be larger. As the map is used, values near C_b will stand for regions whose occupancy state is essentially unknown, while those much nearer zero will

represent empty places, and those much nearer unity are likely to be occupied. Most of the planning algorithms that use the grid will be better off if they do not make sharp distinctions, but instead numerically combine the certainty values from various cells to produce "goodness of fit" numbers for their various hypotheses. In this way the essential uncertainties in the measurements are not masked, and the algorithms do not jump to unnecessary, possibly false, conclusions.

4 Inserting Measurements

The readings of almost any kind of sensor can be incorporated into a certainty grid, if they can be expressed in geometric terms. The information from a reading can be as minimal as a proximity detector's report that there is probably something in a certain region of space, or as detailed as a stereo depth profiler's precise numbers on the contours of a surface.

The first step, in general, is to express the sensor's measurement as a numerical spatial certainty distribution commensurate with the grid's geometry. For an infrared proximity detector this may take the form of a set of numbers P_x in an elliptical envelope with high certainty values in a central axis (meaning detection is likely there) tapering to zero at the edges of the illumination envelope. Let's suppose the sensor returns a binary indication that there is or is not something in its field of view. If the sensor reports a hit, cells in the certainty grid C_x falling under the sensor's envelope can be updated with the formula

$$C_x := C_x + P_x - C_x \times P_x$$

which will increase the C values. In this case the P values should be scaled so their sum is one, since the measurement describes a situation where there is something somewhere in the field of view, probably not everywhere. If the reliability of the sensor is less than perfect, the normalization may be to a sum less than unity. If, on the other hand, the detector registers no hit, the formula might be

$$C_x := C_x \times (1 - P_x)$$

and the C s will be reduced. In this case the measurement states that there is nothing anywhere in the field of view, and the P values should reflect only the chance that an object has been overlooked at each particular position; i.e. they should not be normalized. If the sensor returns a continuous value rather than a binary one, perhaps expressing some kind of rough range estimate, a mixed strategy similar to the one described below for sonar is called for.

A Polaroid sonar measurement is a number giving the range of the nearest object within an approximately thirty degree cone in front of the sonar transducer. Because of the wide angle, the object position is known only to be somewhere on a certain surface. This range surface can be handled in the same manner as the sensitivity distribution of a proximity detector "hit" above. The sonar measurement has something else to say, however. The volume of the cone up to the range reading is probably empty, else a smaller range would have been returned. The empty volume is like the "no hit" proximity detector case, and can be handled in the same fashion. So a sonar reading is like a proximity detector hit at some locations, and increases the occupancy probability there, and like a miss at others, where it decreases the probability. If we have a large number of sonar readings taken from different vantage points (say as the robot moves), the gradual accumulation of such certainty numbers will build a respectable map. We can, in fact, do a little better than that. Imagine two sonar readings whose volumes intersect. And suppose the "empty" region of the second overlaps part of the range surface of the first. Now the range surface says

"somewhere along here there is an object", while the empty volume says "there is no object here". The second reading can be used to reduce the uncertainty in the position of the object located by the first reading by decreasing the probability in the area of the overlap, and correspondingly increasing it in the rest of the range surface. This can be accomplished by reducing the range surface certainties R_x with the formula $R_x := R_x \times (1 - E_x)$ where E_x is the "empty" certainty at each point from the second reading, then normalizing the R s. This method is used to good effect in the existing sonar navigation programs, with the elaboration that the E s of many readings are first accumulated, and then used to condense the R s of the same readings. (It is this two stage process that led us to use two grids in our original programs. In fact, the grid in which the E s are accumulated need merely be temporary working space.)

The stereo method of Serey and Matthies provides a depth profile of visible surfaces. Although, like a sonar reading, it describes a volume of emptiness bounded by a surface whose distance has been measured, it differs by providing a high certainty that there is matter at each point along the range surface. The processing of the "empty" volume is the same, but the certainty reduction and normalization steps we apply to sonar range surfaces are thus not appropriate. The grid cells along a very tight distribution around the range surface should simply be increased in value according to the "hit" formula. The magnitude and spread of the distribution should vary according to the confidence of the stereo match at each point. The method used by Serey and Matthies matches edge crossing along corresponding scanlines of two images, and is likely to be accurate at those points. Elsewhere it interpolates, and the expected accuracy declines.

If the robot has proximity or contact sensors, its own motion can contribute to a certainty grid. Areas traversed by the robot are almost certainly empty, and their cells can be reduced by the "no hit" formula, applied over a confident sharp edged distribution in the shape of the robot. This approach becomes more interesting if the robot's motion has inherent uncertainties and inaccuracies. If the certainty grid is maintained so it is accurate with respect to the robot's present position (so called robot co-ordinates), then the past positions of the robot will be uncertain in this co-ordinate system. This can be expressed by *blurring* the certainty grid accumulated from previous readings in a certain way after each move, to reflect the uncertainty in that move. New readings are inserted without blur (essentially the robot is saying "I know *exactly* where I am now; I'm just not sure where I was before). The track in the certainty grid of a moving robot's path in this system will resemble the vapor trail of a high flying jet - tight and dense in the vicinity of the robot, diffusing eventually to nothing with time and distance.

5 Extracting Deductions

The purpose of maintaining a certainty grid in the robot is to plan and monitor actions. Thorne and Elfes showed the way to plan obstacle avoiding paths. Conceptually the grid can be considered an array of topographic values - high occupancy certainties are hills while low certainties are valleys. A safe path follows valleys, like running water. A relaxation algorithm can perturb portions of a trial path to bring each part to a local minimum. In principle a decision need never be made as to which locations are actually empty and which are occupied, though perhaps the program should stop if the best path climbs beyond some threshold "altitude". If the robot's sensors continue to operate and update the grid as the path is executed, impasses will become obvious as proximity and contact sensors raise the occupancy certainty of locations where they make contact with solid matter.

As indicated in the introduction, we have already demonstrated effective navigation by convolving certainty grids of given locations built at different times, allowing the robot to determine its location with respect to previously constructed maps. This technique can be extended to subparts of maps, and may be suitable for recognizing particular landmarks and objects. For instance, we are presently developing a wall tracker that fits a least squares line to points that are weighted by the product of the occupancy certainty value and a gaussian of the distance of the grid points from an a-priori guess of the wall location. The parameters of the least squares line are the found wall location, and serve, after being transformed for robot motion, as the initial guess for the next iteration of the process.

For tasks that would benefit from an opportunistic exploration of unknown terrain, the certainty grid can be examined to find interesting places to go next. Unknown regions are those whose certainty values are near the background certainty C_b . By applying an operator that computes a function such as

$$\sum (C_x - C_b)^2$$

over a weighted window of suitable size, a program can find regions whose contents are relatively unknown, and head for them. Other operators similar in spirit can measure other properties of the space and the robot's state of knowledge about it. Hard edged characterizations of the stuff in the space can be left to the last possible moment by this approach, or avoided altogether.

6 A Plan: Awareness for a Robot

Uranus is the CMU Mobile Robot Lab's latest and best robot and the third and last one we intend to construct for the foreseeable future. About 60 cm square, with an omnidirectional drive system intended primarily for indoor work, Uranus carries two racks wired for the industry standard VME computer bus, and can be upgraded with off the shelf processors, memory and input output boards. In the last few years the speed and memory available on single boards has begun to match that available in our mainframe computers. This removes the main arguments for operating the

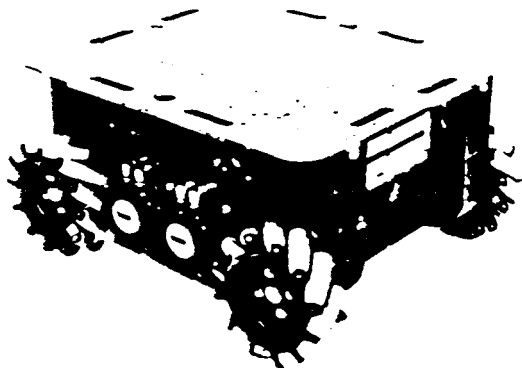


Figure 5: The Uranus Mobile Robot - A bouncing baby, full of promise.

machine primarily by remote control. With most computing done on board by dedicated processors, enabling very high bandwidth and reliable connection of processors to sensors and effectors, real time control is much easier. Also favoring this change in approach is a realization by us, growing from our experience with robot control programs from the very complex to the relatively simple, that the most complicated programs are probably not the most effective way to learn about programming robots. Very complex programs are slow, limiting the number of experiments possible in any given time, and they involve too many simultaneous variables, whose effects can be hard to separate. A manageable intermediate complexity seems likely to get us to our long term goals fastest. The most exciting element in our current plans is a realization that certainty grids are a powerful and efficient unifying solution for sensor fusion, motion planning, landmark identification, and many other central problems.

As the core of the robot and the research we will prepare a kind of operating system based on the "certainty grid" idea. Software running continuously on processors onboard Uranus will maintain a probabilistic, geometric map of the robot's surroundings as it moves. The certainty grid representation will allow this map to be incrementally updated in a uniform way from various sources including sonar, stereo vision, proximity and contact sensors. The approach can correctly model the fuzziness of each reading, while at the same time combining multiple measurements to produce sharper map features, and it can deal correctly with uncertainties in the robot's motion. The map will be used by planning programs to choose clear paths, identify locations (by correlating maps), identify well known and insufficiently sensed terrain, and perhaps identify objects by shape. To obtain both adequate resolution of nearby areas and sufficient coverage for longer range planning, without excessive cost, a hierarchy of maps will be kept, the smallest covering a 2 meter area at 5.25 cm resolution, the largest 16 meters at 50 cm resolution (Figure 6). This map will be "scrolled" to keep the robot centered as it moves, but rotations of the robot will be handled by changing elements of a matrix the represents the robot's orientation in the grid. The map forms a kind of consciousness of the world surrounding the robot - reasoning about the world would actually be done by computations in the map. It might be interesting to take one more step in the hierarchy, to a one meter grid that simply covers the robot's own extent. It would be natural to keep this final grid oriented with respect to robot chassis itself, rather than approximately to the compass as with the other grids. This change of co-ordinate system would provide a natural distinction between "world" awareness and "body" or "self" awareness. Such encoding of a sense of self might even be useful if the robot were covered with many sensors, or perhaps were equipped with manipulators. We have no immediate plans in that direction, and so will pass by this interesting idea for now.

Our initial version will contain a pair of two dimensional grid sets, one mapping the presence of objects at the robot's operating height of a few feet above ground level. The other will map the less complex idea of presence of passable floor at various locations. The object map will be updated from all sensors, the floor map primarily from downward looking proximity detectors, though possibly also from long range data from vision and sonar. The robot will navigate by dead reckoning, integrating the motion of its wheels. This method accumulates error rapidly, and this uncertainty will be reflected in the maps by a repeated blurring operation. Old readings, whose location relative to the robot's present position and orientation are known with decreasing precision, will have their effect gradually diffused by this operation, until they eventually evaporate to the background certainty value.

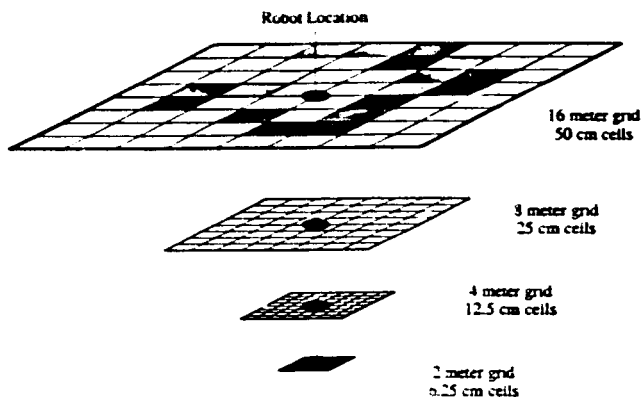


Figure 6: Map Resolution Hierarchy - Coarse maps for the big picture, fine ones for the fiddly details in the immediate environment. All the maps are scrolled to keep the robot in the center cells.

It would be natural to extend the two-grid system to many grids, each mapping a particular vertical slice, until we have a true three dimensional grid. We will do this as our research results, and processing power permit. The availability of single board array processors that can be installed on the robot would help this, as the certainty grid operations are very amenable to vectorizing. The certainty grid representation can also be extended in the time dimension, with past certainty grids being saved at regular intervals, like frames in a movie film, and registered to the robot's current co-ordinates (and blurred for motion uncertainties). Line operators applied across the time dimension could detect and track moving objects, and give the robot a sense of time as well as space. This has some very thrilling conceptual (and perceptual) consequences, but we may not get to it for a while.

Even the simplest versions of the idea will allow us fairly straightforwardly to program the robot for tasks that have hitherto been out of reach. We look forward to a program that can explore a region and return to its starting place, using map "snapshots" from its outbound journey to find its way back, even in the presence of disturbances of its motion and occasional changes in the terrain. By funneling the sensor readings through a certainty grid, which collects and preserves all the essential data, and indications of uncertainties, and makes it available in a uniform way, we avoid the problem we've had, that for each combination of sensor and task a different program is required. Now the task execution is decoupled from the sensing, and thus becomes simpler.

7 References

1. Serey, B. and L. H. Mathies, **Obstacle Avoidance using 1-D Stereo Vision**, CMU Robotics Institute Report, November, 1986.
2. Elfes, A. E. **A Sonar-Based Mapping and Navigation System**, Workshop on Robotics, Oak Ridge National Lab, Oak Ridge, TN, August, 1985. (invited presentation), in the proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, April 7-10 1986 also to appear as an invited paper in IEEE Transactions on Robotics and Automation.
3. Kadonoff, M., F. Benayad-Cherif, A. Franklin, J. Maddox, L. Muller, B. Sert and H. Moravec. **Arbitration of Multiple Control Strategies for Mobile Robots**, SPIE conference on Advances in Intelligent Robotics Systems, Cambridge, Massachusetts, October 26-31, 1986. In SPIE Proceedings Vol 727, paper 727-10.
4. Moravec, H. P. and A. E. Elfes, **High Resolution Maps from Wide Angle Sonar**, proceeding of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, March, 1985, pp 116-121, and proceedings of the 1985 ASME conference on Computers in Engineering, Boston, August, 1985.
5. Thorpe, C. E., **Path Relaxation: Path Planning for a Mobile Robot**, CMU-RI-TR-84-5, Robotics Institute, Carnegie-Mellon University, April, 1984. also in proceedings of IEEE Oceans 84, Washington, D.C., August, 1984 and Proceedings of AAAI-84, Austin, Texas, August 6-10, 1984, pp. 318-321.
6. Moravec, H. P., **Robot Rover Visual Navigation**, UMI Research Press, Ann Arbor, Michigan, 1981. also available as **Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover**, Stanford AIM-340, CS-80-313 and CMU-RI-TR-3.