Topological mapping for mobile robots using a combination of sonar and vision sensing

David Kortenkamp*and Terry Weymouth
Artificial Intelligence Laboratory
The University of Michigan
Ann Arbor, MI 48109
korten@aio.jsc.nasa.gov

Abstract

Topological maps represent the world as a network of nodes and arcs: the nodes are distinctive places in the environment and the arcs represent paths between places. A significant issue in building topological maps is defining distinctive places. Most previous work in topological mapping has concentrated on using sonar sensors to define distinctive places. However, sonar sensors are limited in range and angular resolution, which can make it difficult to distinguish between different distinctive places. Our approach combines a sonar-based definition of distinctive places with visual information. We use the robot's sonar sensors to determine where to capture images and use cues extracted from those images to help perform place recognition. Information from these two sensing modalities is combined using a simple Bayesian network. Results described in this paper show that our robot is able to perform place recognition without having to move through a sequence of places, as is the case with most currently implemented systems.

Introduction

Topological maps represent the world as a graph of places with the arcs of the graph representing movements between places. Brooks (Brooks 1985) argues persuasively for the use of topological maps as a means of dealing with uncertainty in mobile robot navigation. Indeed, the idea of a map that contains no metric or geometric information, but only the notions of proximity and order, is enticing because such an approach eliminates the inevitable problems of dealing with movement uncertainty in mobile robots. Movement errors do not accumulate globally in topological maps as they do in maps with a global coordinate system since the robot only navigates locally, between places. Topological maps are also much more compact in their representation of space, in that they represent only certain places and not the entire world, in contrast to robots which use detailed *a priori* models of the world, such as (Kosaka & Kak 1992) and (Fennema & Hanson 1990). For these reasons, topological maps have become increasingly popular in mobile robotics.

A significant issue in building a topological map is defining distinctive places in the environment; these distinctive places correspond to the nodes of the resulting topological map. Most researchers use sonar sensors to define distinctive places (Basye, Dean, & Vitter 1989; Kuipers & Byun 1991; Mataric 1992). However, sonar sensors are limited in range and angular resolution and therefore can only give a rough approximation of the robot's environment. Because of this, many "distinctive" places in the environment actually look very similar to sonar sensors. For example, in a long hallway with left and right doorways to rooms, using only sonar sensors it would be impossible to distinguish any particular left or right door from any other left or right door along the hallway. Most systems overcome this limitation by determining the robot's location based on a *sequence* of distinctive places instead of on a single distinctive place. Such approaches, while certainly effective, require the robot to make many navigational movements in order to determine its location in the environment.

It is our hypothesis that by adding visual information to the robot's sonar information, we can dramatically reduce the ambiguity of places that look identical to the robot's sonar sensors. In our approach, sonar sensors are used to determine generic places in the environment called *gateways*. Gateways mark the transition from one space to another space. Since a gateway marks the entrance to a new space, they offer the robot a perfect opportunity to look around and acquire visual cues that will distinguish among gateways. Thus, at each gateway one or more images (*scenes*) are captured. Visual cues are extracted from the image and stored with the gateway. On subsequent visits to the same gateway, the robot can use the visual cues, in conjunction with the sonar signature of the gateway, to determine its location.

Sonar information

Most topological maps are built around distinctive places. In our topological map, rather than looking

*Now at The MITRE Corporation, Houston, TX 77058.
This research was sponsored by Department of Energy grant DE-FG02-86NE37969

for places that are locally distinguishable from other places and then storing the distinguishing features of the place in the route map, we instead look for places that mark the transition between one space in the environment and another space. We call these places *gateways*.

In indoor environments, gateways are places such as entrances to rooms and intersections of hallways. For a mobile robot in an indoor environment, gateways are important for several reasons. First, gateways tend to be places that are visited frequently. Second, gateways are places that open up new views for a robot, views from which it can extract visual cues to distinguish between similar gateways. Third, a robot typically must go through a gateway in a small number of directions. For example a robot can only pass through a doorway in two directions. This constrains the range of views that a robot can have at a gateway and simplifies matching of visual cues. Finally, gateways are exits from a space and, for safety reasons, a robot should stay aware of exits.

Detecting gateways

We have defined gateways, for orthogonal indoor environments, as openings to the left or right of the robot's direction of travel that are wide enough for the robot to pass through. These openings are detected using sonar sensors. Our gateway detection algorithm has the following components:

1. The robot aligns itself along a wall (or along both walls of a corridor) using its sonar sensors.
2. The robot moves along the wall and maintains its orientation and distance with respect to the wall (or walls in a corridor) using its sonar sensors.
3. While moving, the robot continually checks its left and right sonar readings for openings; the robot also checks for obstacles in front of it.
4. When an opening is found, the robot continues moving and looks for a closing to that opening. While looking for a closing, the robot also checks the opposite direction for any openings as well as checking the front for any obstacles.
5. When a closing to the opening is found (a closing can be an end to the opening, a blockage in front, or a certain distance traveled), the robot determines if the opening is large enough to pass through and, if so, signals a gateway.
6. The robot positions itself in the middle of the gateway.

Experiments with our Labmate TRC robot in the hallways of our laboratory show that this gateway detection algorithm has an error of no more than 3.5 degrees in orientation along the axis of the hallway and 70mm in position along the axis of the hallway. These errors were determined by repeatedly having the robot stop at the same set of gateways and measuring its orientation and location.

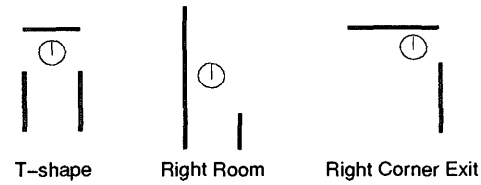

Figure 1: A few examples of different types of gateways.

Classifying gateways

Once a gateway has been detected it can be classified as a certain type using local sensory information. For example, a T-SHAPE gateway is characterized by a simultaneous opening on both the left and right of the robot followed by the robot being blocked in the front (see figure 1). A RIGHT ROOM gateway is characterized by a right opening followed by no closing. In rooms, gateways are typically exits, such as the RIGHT CORNER EXIT in figure 1. We have identified a total of 25 gateway types in typical indoor environments. Classifying each gateway using local sensory information helps the robot perform place recognition.

Gateways extend the traditional sonar-based topological place by being not just distinctive places, but *important* places in that they open up new views for the robot. The robot can take advantage of these views to store visual scenes that can help it distinguish between gateways. In essence, gateways represent *generic* places in the environment (for example, doors, intersections) and not *specific* places (for example, the door to room 200). This is acceptable for some forms of navigation. For example, if the robot is told to take the third right opening, then the algorithms described in this section will be perfectly adequate. However, if the robot does not know its starting location or the environment changes (that is, the second opening on the right is closed) then simply relying on local sonar information can be dangerous. For this reason, our gateway mechanism is augmented with visual information, described in the following section.

Visual information

We augment our sonar information with visual information. Our visual information takes the form of visual scenes captured at gateways from which we extract visual cues. Scenes, as they are presented here, differ from the traditional computer vision paradigm. In our approach, cues are not simply extracted from an image and then stored apart from the scene, but their *location* in the scene is of equal importance. Kaplan, Kaplan, and Lesperance (Kaplan 1970; Kaplan & Kaplan 1982;

Lesperance 1990) discuss the importance of a fast and unobtrusive mechanism that gives a rough assessment of the objects surrounding an organism and their relationships to each other.

Extracting visual cues

Programming a robot to autonomously find visual cues (or landmarks, although that term is generally used for highly complex objects as opposed to the simple features discussed in this subsection) is an area of active research and there are several proposed landmark detection algorithms (Levitt & Lawton 1990; Tsuji & Li 1993). For our experiments we have chosen a simple cue—vertical edges. Vertical edges have proven very useful as indoor visual cues in other mobile robot systems (Kriegman, Triendl, & Binford 1989; Crowley *et al.* 1991) and are especially effective in our experimental space due to the sharp contrast between black doorway frames and white walls. Vertical edges are extracted from a black-and-white image by a modified Sobel edge detector. A second image is analyzed in the same way as the first, but it is offset by 18cm from the first image. This produces two lists of edges, one list for the right image and one list for the left image. The edges on the two lists are matched to each other using the direction of transition of the edge (that is, was the edge from light to dark or dark to light?), length, and location. The pixel shift in the matched edges from the first image to the second image (called the disparity) is calculated and used to determine a rough distance to the edge. Each visual cue, thus, has three scene-independent features: direction, length, and distance.

Storing visual scenes

We store the robot's visual cues in an abstracted scene representation (ASR), which is a 5 X 5 grid. The choice of a 5 X 5 grid size is based on the experiments in sonar gateway detection. The orientation error at a gateway is a maximum of 3.5 degrees. Using a camera with a focal length of 4.8mm, a 3.5 degree variation in orientation yields a 47 pixel displacement for cues 2m away (objects further away will have a smaller disparity). Doubling this to 94 pixels and dividing it into the image size of 480 X 480 pixels gives a 5 X 5 abstracted grid. In such a grid, a cue that falls in the middle of a cell and is further than two meters away will remain in that same cell given a 3.5 degree difference in orientation. Each cell of the ASR can be connected to a representation of a visual cue that occupies that location in the scene. In the current system, the representation of the visual cue contains the direction, distance, and length of the cue. However, in more sophisticated implementations the representation of the visual cue could contain detailed information about how to recognize the cue, maybe even a small neural network that performs pattern recognition for the pattern located in those cells.

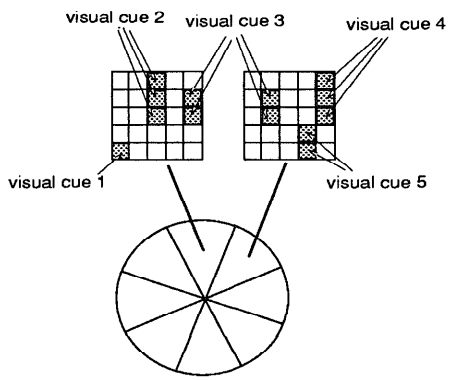

Figure 2: Cues are stored in abstracted representations.

ASRs can be stored for any of eight directions that the robot can be told to face at a gateway. The number of directions represented is dependent on the visual field of view; in our robot the field of view is 60 degrees, which allowing for overlap, gives 8 directions each representing 45 degrees. Typically, the robot will only store one scene (in the forward direction) at each gateway. Figure 2 shows some sample ASRs and how several can be stored at a single gateway.

ASRs are not static structures as that would render them useless in a dynamic world. An ASR should only contain those cues that remain constant over many traversals, since they will be the most reliable cues in the future. To accomplish this, the connection that links each cell of the ASR to a cue representation can be strengthened or weakened, depending on whether the cue is present or not during each traversal.

Currently, it is necessary to have guided training runs in order to build up a stable set of cues. In the future, we would like our robot to explore autonomously, attempting to determine where it is as well as it can and updating appropriate maps as much as it can. Such a system would make more mistakes at first, but would not need to rely on directions to find routes to goals. Also, such a system would take much longer to learn a stable set of cues. The current implementation can be compared to someone taking you on a guided tour of a building several times before letting you loose. During training the robot is only told at which gateway it currently is. The robot detects and stops at the gateways completely autonomously.

Matching ASRs

The robot must have some mechanism for matching its current scene with the ASRs that are stored with each gateway. Four different match algorithms were implemented and then compared using actual scenes acquired by the robot. The four algorithms are: 1) a feature-to-feature match using distance and direction

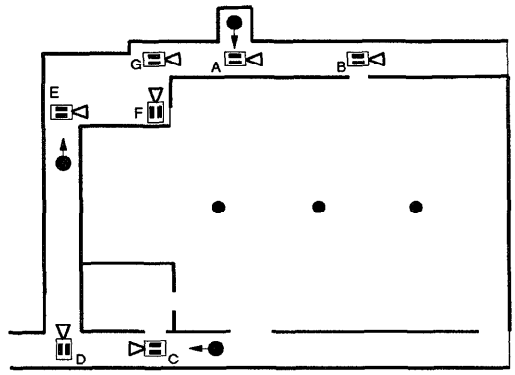

Figure 3: The experimental space for place recognition.

in which an entire feature in the current scene must match an entire feature in the stored scene; 2) a cell-to-cell match using distance and direction in which each occupied cell in the current scene must match an occupied cell in the stored scene; 3) a cell-to-cell match using only direction; 4) a cell-to-cell match using only occupancy.

The comparison procedure consists of having the robot traverse three routes five times each and building up 16 ASRs at seven different gateways. Figure 3 shows the locations of the gateways. In this figure, the black circles with arrows are the three starting points for the routes. After the robot has acquired its ASRs, it traverses each route a final time; this is the testing run. It then matches each scene along the testing run with all of the ASRs stored during the initial traversals. A current scene was said to match a stored ASR if the ratio of matched cues (or cells in methods 2, 3, and 4) to total stored cues (or cells) is higher than any other scene (a tie resulted in no match). Under these conditions, method 1 matched seven out of sixteen scenes, method 2 matched nine of sixteen, method 3 matched five of sixteen, and method 4 matched six of sixteen. Given these results, the second match algorithm was chosen for the remainder of the experiments in this paper.

Our notion of storing visual scenes to aid mobile robot navigation is not unique; there has been active research in using visual scenes to provide robots with *homing* capabilities. These robots do not build maps, but instead the robot stores sensory data about the environment and associates movements with sensory events. As sensory events trigger movements, the robot navigates the environment. Examples of homing robots are (Nelson 1989) and (Hong *et al.* 1992). Our contribution is that instead of storing visual scenes at regular intervals, as is done in homing, we store visual scenes only at locations that are considered interesting by the robot's sonar sensors.

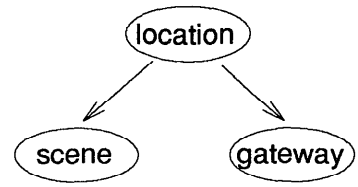

Figure 4: The Bayesian network used for place recognition

	Stored places						
	A	B	C	D	E	F	G
A	.43	.09	.22	.05	.05	.1	.06
B	.05	.52	.21	.06	.05	.05	.05
C	.10	.12	.36	.20	.04	.13	.04
D	.14	.05	.24	.43	.05	.04	.05
E	.14	.14	.14	.14	.14	.14	.14
F	.14	.14	.14	.16	.14	.14	.14
G	.14	.14	.14	.14	.14	.14	.14

Table 1: Likelihoods for each place using only vision.

Place recognition

A single source of information, whether it be sonar or vision, is not enough to perform robust place recognition without further navigation. Thus, we use both gateway characterization and visual cues in the place recognition process and combine them using a simple Bayesian network (Pearl 1988), in which a location is determined by a scene and a gateway. The probabilistic network used by our robot is shown in Figure 4. The scene node of our network is the likelihood of a given location as determined by matching the visual cues stored for that location with the cues in the current visual scene. The gateway node is the likelihood of a given location determined by comparing its classification with the classification of the current gateway. Both of these leaf nodes are combined to determine the robot's location.

The goal of integrated place recognition is to perform place recognition better using a combination of vision and sonar than would be possible using either alone. Better is defined in three ways: 1) A higher accuracy in place recognition; 2) A greater resilience to sensor errors; and 3) An ability to resolve ambiguous places.

We have tested our topological mapping system using a real robot in an unaltered environment. In our experimental set-up, the robot has built up ASRs of seven places along three routes each traversed five times (see figure 3). Each ASR has one scene in the forward direction and has a gateway classification for each place. Then the three routes are traversed a final time and a test scene is stored at each of the seven gateways, along with a test gateway classification. The test scene is matched against all the stored ASRs and the test classification is matched with all the stored classifications

	Stored places						
	A	B	C	D	E	F	G
A	.82	.04	.04	.04	.04	0	0
B	.02	.31	.31	.31	.06	0	0
C	.02	.31	.31	.31	.06	0	0
D	.02	.31	.31	.31	.06	0	0
E	.04	.12	.12	.12	.61	0	0
F	0	0	0	0	0	.90	.10
G	0	0	0	0	0	.10	.90

Table 2: Likelihoods for each place using only sonar.

	Stored places						
	A	B	C	D	E	F	G
A	**.95**	.01	.02	.01	.01	0	0
B	0	**.65**	.26	.07	.01	0	0
C	0	.17	**.52**	.29	.01	0	0
D	.01	.07	.33	**.58**	.01	0	0
E	.04	.12	.12	.12	**.61**	0	0
F	0	0	0	0	0	**.90**	.10
G	0	0	0	0	0	.09	**.91**

Table 3: Combined likelihoods (vision and sonar) for each place.

in order to do place recognition.

Table 1 gives the likelihood for each place using only the visual evidence. Across the top are the stored places and down the side are the places at which the robot is (that is, the test scenes). The numbers reflect the likelihood that the robot is at that place given the visual scene information. They were determined by normalizing the percentage of matching cues between the test scene and the stored ASRs. For example, if there are three scenes and the match percentages are: .25, .90, and .75 then the likelihoods would be .13, .47, and .40. If no cues matched then the algorithm still assigned a small match percentage (0.10) to that place, since a zero likelihood would cause the final likelihood for that place to be zero no matter how strong the sonar evidence. The table shows that four out of seven places (A,B,C, and D) would be correctly identified (that is, have the highest likelihood) using only visual information.

Table 2 gives the likelihoods for each place using only sonar information (that is, gateway characterization). These likelihoods were determined by us and entered into the system: the correct gateway is given the highest likelihood; similar gateways are given much smaller likelihoods; and dissimilar gateways are given a zero likelihood. Sonar information also gives a 57% accuracy in place recognition. This is because three places (B, C, and D) look identical to the sonar sensors and are all characterized as RIGHT OPENING, so only four out of the seven places (57%) can be uniquely recognized (that is, have the highest likelihood) using sonar sensors.

Finally, Table 3 shows the likelihoods for each place

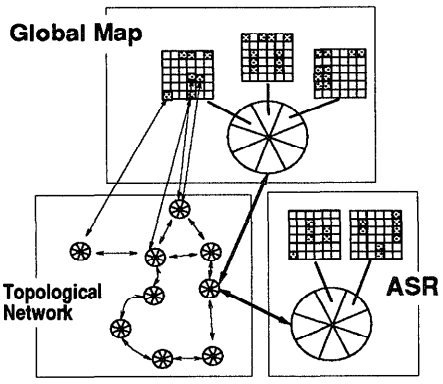

Figure 5: Each ASR is a small component of a larger representation.

when the vision and sonar evidence is combined using the Bayesian network (both sources of evidence are weighted equally). While consisting of a small number of places, this experiment demonstrates how sonar and vision combined can result in place recognition that is more accurate than would be possible using either sensing modality by itself; when the sonar information is ambiguous, the visual evidence distinguishes between places, and vice versa.

Integrating sonar and vision can also help overcome sensor errors during place recognition. For example, let's assume that the robot misclassified place D as a RIGHT ROOM instead of a RIGHT OPENING (in reality the robot never made this mistake, so the error had to be simulated). In this case, its sonar likelihood for place D is only .11, while its sonar likelihood for place E (which actually is a RIGHT ROOM) is .61. When vision evidence is considered, the likelihoods are updated to .42 for place D and only .25 for place E, thus correcting the sonar error.

Conclusion

All previous research into topological mapping for mobile robots uses sonar sensing for place recognition. Many mobile robots also use vision sensing for place recognition. Both approaches have their merits and we believe that combining sonar and vision sensing in a topological representations results in a better robot navigation system. Our system can reduce or eliminate the need for additional robot movements to distinguish between places that appear identical to sonar sensors and it can also reduce the number of scenes that need to be stored by only acquiring scenes at those places that are determined as interesting by the sonar sensors. There are also some drawbacks to our approach when compared to other systems. First, our robot requires several initial, guided traversals of a route in order to acquire a stable set of locational cues to navigate autonomously. Second, acquiring, storing and matching

visual scenes is very expensive, both in computation and storage. Finally, we are restricted to highly structured, orthogonal environments.

There is also the question of how our system will scale up, given that our experimental space consisted of only seven gateways, due to the time consuming nature of experimenting with real robots. Certainly, the perfect place recognition performance we achieved in our experiments will not hold up as more and more places are added. However, it is unrealistic to expect the robot to have no idea of where it has started; this is a worst case scenario used for experimental purposes only. As the robot gets more gateways we expect that knowledge of the robot's previous location can eliminate all but a handful of possibilities for the current location, which can then be resolved using sensory information as was demonstrated in our experiments.

In the future we hope to expand our robot's visual sensing beyond simple vertical edges. We are also in the process of implementing a better gateway detection algorithm that incorporates more sophisticated obstacle avoidance (see (Kortenkamp *et al.* 1994) for preliminary results). On a broader scale, this work is a small part of a larger robot mapping system detailed in (Kortenkamp 1993). The larger system address such issues as representing the topological map, extracting routes from the topological map, traversing previously learned routes and building a geometric map from the topological data. The complete representation is shown in figure 5. Each ASR is a node in the topological network upon which a global map is constructed. The global map has a structure similar to an ASR but instead of storing visual cues it stores locations of distant places. This representation is based on a cognitive model of human spatial mapping described in (Chown, Kaplan, & Kortenkamp 1994).

References

Basye, K.; Dean, T.; and Vitter, J. S. 1989. Coping with uncertainty in map learning. In *Proceedings of the International Joint Conferences on Artificial Intelligence.*

Brooks, R. A. 1985. Visual map making for a mobile robot. In *Proceedings IEEE Conference on Robotics and Automation.*

Chown, E.; Kaplan, S.; and Kortenkamp, D. 1994. Prototypes, location and associative networks (PLAN): Towards a unified theory of cognitive mapping. To appear in *The Journal of Cognitive Science.*

Crowley, J. L.; Bobet, P.; Sarachik, K.; Mely, S.; and Kurek, M. 1991. Mobile robot perception using vertical line stereo. *Robotics and Autonomous Systems* 7(2-3).

Fennema, C., and Hanson, A. R. 1990. Experiments in autonomous navigation. In *Proceedings Image Understanding Workshop.*

Hong, J.-W.; Tan, X.; Pinette, B.; Weiss, R.; and Riseman, E. M. 1992. Image-based homing. *IEEE Control Systems* 12(1):38–45.

Kaplan, S., and Kaplan, R. 1982. *Cognition and Environment: Functioning in an Uncertain World.* Ann Arbor, MI: Ulrichs.

Kaplan, S. 1970. The role of location processing in the perception of the environment. In *Proceedings of the Second Annual Environmental Design Research Association Conference.*

Kortenkamp, D.; Huber, M.; Koss, F.; Lee, J.; Wu, A.; Belding, W.; and Rogers, S. 1994. Mobile robot exploration and navigation of indoor spaces using sonar and vision. In *Proceedings of the AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space (CIRFFSS '94).*

Kortenkamp, D. 1993. *Cognitive maps for mobile robots: A representation for mapping and navigation.* Ph.D. Dissertation, The University of Michigan.

Kosaka, A., and Kak, A. C. 1992. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *Computer Vision, Graphics, and Image Processing* 56(2).

Kriegman, D. J.; Triendl, E.; and Binford, T. O. 1989. Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation* 5(6).

Kuipers, B. J., and Byun, Y.-T. 1991. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems* 8.

Lesperance, R. 1990. *The Location System: Using Approximate Location and Size Information for Scene Segmentation.* Ph.D. Dissertation, The University of Michigan.

Levitt, T. S., and Lawton, D. T. 1990. Qualitative navigation for mobile robots. *Artificial Intelligence* 44(3).

Mataric, M. K. 1992. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation* 8(3).

Nelson, R. C. 1989. Visual homing using an associative memory. In *Proceedings of the Image Understanding Workshop.*

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann.

Tsuji, S., and Li, S. 1993. Memorizing and representing route scenes. In Meyer, J.-A.; Roitblat, H. L.; and Wilson, S. W., eds., *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior.* Cambridge, MA: MIT Press.