From: AAAI-96 Proceedings. Copyright © 1996, AAAI (www.aaai.org). All rights reserved.

Navigation for Everyday Life

Daniel D. Fu and Kristian J. Hammond and Michael J. Swain
Department of Computer Science, University of Chicago
1100 East 58th Street, Chicago, Illinois 60637

Abstract

Past work in navigation has worked toward the goal of producing an accurate map of the environment. While no one can deny the usefulness of such a map, the ideal of producing a complete map becomes unrealistic when an agent is faced with performing real tasks. And yet an agent accomplishing recurring tasks should navigate more efficiently as time goes by. We present a system which integrates navigation, planning, and vision. In this view, navigation supports the needs of a larger system as opposed to being a task in its own right. Whereas previous approaches assume an unknown and unstructured environment, we assume a structured environment whose organization is known, but whose specifics are unknown. The system is endowed with a wide range of visual capabilities as well as search plans for informed exploration of a simulated store constructed from real visual data. We demonstrate the agent finding items while mapping the world. In repeatedly retrieving items, the agent's performance improves as the learned map becomes more useful.

Introduction

Past work in navigation has generally assumed that the purpose of navigation is to endow a robot with the ability to navigate reliably from place to place. However, in focusing on this specific problem, researchers have ignored a more fundamental question: What is the robot's purpose in moving from place to place? Presumably the robot will perform some later-to-be-named task: The point of navigation is only to get the robot to the intended destination. What happens afterwards is not a concern. This notion has led researchers toward building systems whose sensing ultimately relies on the lowest common denominator (e.g., sonar, dead reckoning). We believe that: (1) Robots will use navigation as a store of knowledge in service of tasks, and (2) Robots will have semantically rich perception in order to perform a wide range of tasks. Given these two beliefs we suggest first that: Navigation must coexist with a robot's planning and action mechanisms instead of being a task in and of itself, and second that: Rich perception, used for tasks, can also be used for constructing a map to make route following and place recognition more tractable.

In this paper we present a system which embodies these two notions as they apply to these five areas: run-time planning, context-based vision, passive mapping, path planning, and route following. This system differs from past navigation research in several ways - the principal difference being the integration of a passive mapper with a planning system. This notion was introduced by Engelson and McDermott (1992). They view a mapping system as a resource for a planner: A mapping subsystem maintains a map of the world as the planner accomplishes tasks. When applicable, the planner uses the acquired map information for achieving goals.

In contrast, traditional research has viewed the mapping subsystem as an independent program which explores and maps its environment. Eventually the program produces a complete map. While no one can doubt the usefulness of such a map, we believe that this mapping may be neither realistic nor necessary. Previously, we showed that an agent can instead use its knowledge of a domain's organization in order to accomplish tasks (Fu *et al.* 1995). This was shown to be effective in a man-made domain - a grocery store. Whereas past approaches have assumed an unknown and unstructured environment, e.g. (Elfes 1987), our approach assumes an environment whose structure is known, but whose specifics are unknown.

For common tasks such as shopping in a grocery store, finding a book in a library, or going to an airport gate, there is much known about each setting which allows the average person to achieve his goals without possessing prior knowledge of each specific setting. For example, people know that: managers of grocery stores organize items by their type, how they're used, etc; librarians shelve books according to category; airport architects account for typical needs of travelers by putting signs in relevant areas. People count on these regularities in order to act appropriately and efficiently without first completely mapping the environment. In this sense, the domains are known. Moreover, the domains are actively structured by someone else so we can depend on regularities being maintained.

We can also rely on certain stable physical properties. Previously we showed that grocery stores exhibit several useful physical properties allowing us to build fast and reliable vision sensing routines. For example, light comes from above, shelves always stock items, items are always displayed facing forward, etc. This work is similar to recent research done in the areas of context-based (Strat and Fischler 1991; Firby *et al.* 1995) and lightweight (Horswill 1993) vision. These paradigms have produced perceptual algorithms which compute useful information reasonably fast so long as the reliability conditions for each perceptual algorithm are known. An immediate consequence of these conditions is that at least some of the sensing routines don't have to be run continuously. If these routines are used in the navigation process, the map must represent the different conditions and results of running a sensing routine. In contrast, previous research has often committed to using uniform fixed-cost sensing (e.g., sonar). This commitment allows the same sensing to be used for both map learning and route following. However, since we use conditional sensing routines, the map learning and route-following methods are markedly different.

In summary, the research presented here differs from traditional research in two major ways. First, we view map learning and route-following in the context of a larger system which performs a wide range of tasks in addition to navigation. We present a framework from which existing planners can integrate navigation. Second, we assume a known and structured environment which enables us to write effective search plans as well as to build a wide range of visual capabilities. We describe a method for using these capabilities in the navigation task.

Shopper

In order to study some of the types of knowledge and underlying mechanisms involved in everyday tasks, we selected grocery store shopping. Shopping is a common activity which takes place in a completely man-made environment. Previously, we showed how our system, SHOPPER, used structural knowledge of the environment to quickly find items using a small set of regularities: items of a similar nature are clustered together (e.g., cereals) as well as items which are often used together (e.g., pancake mix and maple syrup). Several regularities were then encoded into search plans for finding items. For example, if SHOPPER is looking for Aunt Jemima's pancake mix and it sees a "syrup" sign at the head of an aisle, it executes a plan to search an aisle for syrup. After locating the syrup, it executes another search plan for finding the pancake mix in the local area close to the syrup.

These plans were tested in GROCERYWORLD; a simulator of a real grocery store. GROCERYWORLD provides range information from eight sonars plus compass information. SHOPPER is cylindrical with sonars placed equidistant along the circumference. SHOPPER also possesses a panning head. Figure 1 shows the GROCERYWORLD user interface. The top left pane shows SHOPPER in the grocery store with both head and body orientations. The top right pane shows the body-relative sonar readings while the bottom pane shows the sign information available to the agent. If the agent is located at the head of an aisle and is facing the aisle, GROCERYWORLD can supply sign data.

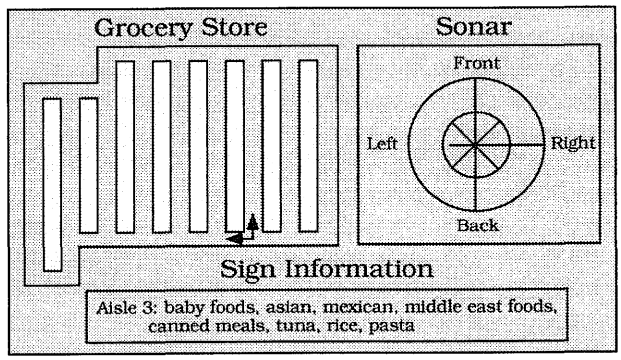

Figure 1: User Interface

GROCERYWORLD differs from most robot simulators in that it supplies real color images taken from a local grocery store. Camera views are restricted to four cardinal directions at each point in the store. Altogether, the domain consists of 75,000 images. Figure 2 shows example views.

GROCERYWORLD also differs from past simulators in that travel is limited to moving along one dimension, except at intersections. However, stores, like office environments, don't have much free space; in fact, hallways and store aisles constrain movement to be in two obvious directions. Kortenkamp and Weymouth (1994) showed that a robot could stay centered in an office hallway with less than 3.5 degrees of orientation error. In light of this fact, we do not believe the one-dimensional travel restriction is a serious shortcoming since we actually prefer SHOPPER to stay centered in an aisle for consistent vision perception.

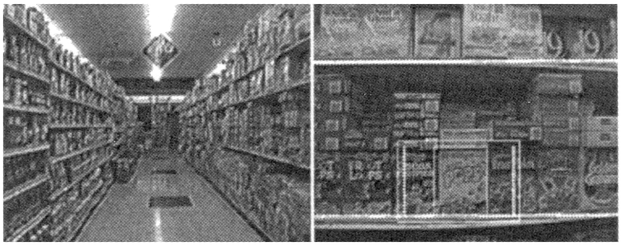

Figure 2: Example views. Left: A typical view down an aisle. Right: A view to the side. Two horizontal lines denote shelf locations. Color regions are enclosed by the larger rectangle. The smaller rectangle around Corn Pops denotes identification.

SHOPPER's perceptual apparatus is primarily suited towards moving around and identifying objects in an image. Below we outline the various sensing routines and explain their use.

Compass: SHOPPER moves and looks in four cardinal directions. We use a compass as an aid to mapping the environment. For example, if SHOPPER knows there is a "soup" sign in view at a particular intersection, it can turn to that direction and attempt to sense the sign.

Sonar: We have sonar sensing continuously for classifying intersections and verifying proximity to shelves.

Signs: When SHOPPER is looking down an aisle and attempts to detect signs, GROCERYWORLD supplies the text of the signs. In Figure 2 a diamond-like sign can be seen above the aisle. However, the image resolution, sign transparency, and specularity prohibit any useful reading.

Shelf Detection: This sensor finds the vertical location of steep gradient changes in an image by smoothing and thresholding for large gradients.

Color Histogram Intersection: Sample color histograms (Swain and Ballard 1991) are taken successively above a shelf and compared to a sought object's histogram in order to identify potential regions according to the intersection response.

Edge Template Matcher: Given an image of object, we use an edge image template matching routine using the Hausdorff distance (Rucklidge 1994) as a similarity metric. This sensor is the most expensive, so it processes areas first filtered by the shelf and color histogram detectors.

All of the above vision algorithms have been implemented and are used by SHOPPER.

Navigation

SHOPPER's navigation system is comprised of four subsystems: RETRIEVER, PATH PLANNER, FOLLOWER, and PASSIVE MAPPER. These subsystems are shown in Figure 3. Initially a goal item is given to the RETRIEVER which uses similarity metrics and the current map to select a destination and a search plan to execute on arrival. The PATH PLANNER replaces the destination with a sequence of nodes (created on previous visits) leading from the current location to the destination. The nodes denote intersections and places of interest. Each node is annotated with accumulated sensor information associated from past visits. Next, the FOLLOWER uses the sequence to follow the path. Vision algorithms are selected and run to ensure correspondence between the predicted sensor information and the current perception. Once the path has been followed, the PLAN INTERPRETER executes the search plan. The PASSIVE MAPPER updates the map during the time the search plan is executed.

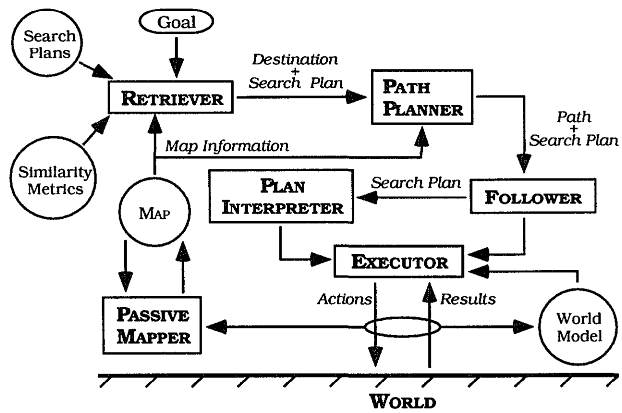

Figure 3: SHOPPER's Architecture. Arrows indicate data flow from one module to another.

In the following sections we describe the four subsystems. Later, using examples, we explain how they interact with each other as well as with the rest of the system.

Passive Mapping

The purpose of the PASSIVE MAPPER subsystem is to maintain a topological map of the world. This subsystem is active when the agent is exploring the world via search plans by monitoring the EXECUTOR as it performs visual and physical actions. The actions and results are used for creating the map. For each physical action the agent performs, it commits, as a policy, to perform a fixed-cost sensing procedure consisting of a compass reading and sonar readings. When the agent knows where it is and exactly where it's going, the PASSIVE MAPPER is disabled since the current map will suffice for route following.

Map Representation. Our map representation draws from previous navigation work using topological maps (Brooks 1985; Engelson and McDermott 1992; Kortenkamp and Weymouth 1994; Kuipers and Byun 1988; Mataric 1992). These maps use relations between places (aka landmarks, distinctive places, gateways, waypoints) for navigation. These methods require the robot be able to recognize a place, and travel from place to place.

We use a topological map consisting of distinctive places and connecting edges. In Kuipers and Byun's NX model, distinctive places are local maxima according to a geometric criterion. Examples of these can be: beginnings of open space, transitions from different spaces, dead ends, etc. Essentially, a distinctive place is a landmark which a robot can recognize and use for map learning and route following. In contrast, our notion of a distinctive place is closely coupled to a place's usefulness to the agent, either as a navigation point where an agent may move into a different space, or a location that is somehow relevant to the agent. In the GROCERYWORLD domain these places are, respectively, *intersections* (INTER's) and *places of interest* (POI's).

Recall that SHOPPER is constrained to move along one dimension at a time. It can move in another dimension only at intersections. One example of a distinctive place is an intersection. Similar to Kortenkamp and Weymouth, we categorize intersections qualitatively as T-SHAPE, CROSS, and CORNER. These descriptions of space are based only on sonar readings. Examples are illustrated in Figure 4. The other example of a distinctive place is a place of interest. These places denote locations important to the agent's goals. For SHOPPER, each place of interest corresponds to a location where SHOPPER found a sought item.

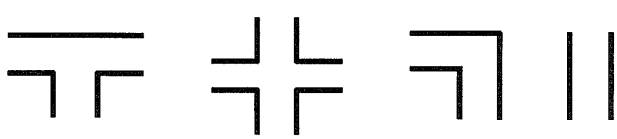

Figure 4: Qualitative descriptions of space: T-SHAPE, CROSS, CORNER, and CORRIDOR.

As SHOPPER encounters new distinctive places, it updates its map by storing perceptual information associated with the distinctive place. The distinctive place description, describing both intersections and places of interest, is a set of tuples $\langle T, S \times C, A, R\rangle$ where:

- $C \in \{0, 1, 2, 3\}$ is a compass direction.
- $T \in$ {T-SHAPE, CROSS, CORNER, CORRIDOR} $\times$ C is the distinctive place type and orientation.
- $S \in$ {sign, shelf, color, template matcher} is a sensing routine. $S \times C$ also accounts for the head's direction at the time the routine was executed.
- A is a set of parameter arguments supplied to the sensing routine.
- R is the output of the routine.

Note that fixed-cost sensors compass and sonar are automatically associated with each distinctive place. For the agent's location in Figure 1, an example sign sensing tuple is: $\langle$(T-SHAPE, 2), (sign, 0), $\emptyset$, { Aisle-3, Baby-foods, Asian, Mexican, Middle-east-foods, Canned-meals, Tuna, Rice, Pasta}$\rangle$.

Landmark Disambiguation. As the PASSIVE MAPPER encounters a new distinctive place, it attempts to determine whether or not it's been there before. For passive mapping, there are two problems to landmark disambiguation: passivity and disparate sensing.

A tension exists between keeping a mapper passive (so as not to interfere with plan execution) and supplying enough information to the mapper for dependable navigation. There are two ways to alleviate this tension:

1. Maintain possible distinctive places. This method, proposed by Engelson and McDermott, requires possibilities to be eventually culled as the agent moves about in the world. Map updates are delayed until the agent has disambiguated its position.

2. Assume rich sensing. The only reason the PASSIVE MAPPER is activated is precisely because it's exploring the environment. If the agent knows where it is, it would be following a route instead. Basically, if SHOPPER doesn't know where it is, assume it will "look around" (Ishiguro *et al.* 1994). Since the PASSIVE MAPPER is active when SHOPPER executes a search plan, we adopt this assumption in the PASSIVE MAPPER as it appears tenable for the time being.

A related problem surfaces when a passive mapper uses a wider range of sensing routines. Since SHOPPER does not run all its sensing routines all the time, we can't guarantee any other sensing routines were run except for the fixed costs of compass and sonar. For

SHOPPER, the plans that are run are search plans. Consequently, we depend on there being additional sensing information as the agent explores the environment. If the agent were executing a different set of plans which involve movement, there might be an altogether different set of sensing routines run. If, however, we commit to changing the agent's location only through search plans or route following, distinctive places will not be confused with each other.

Retriever

Given a goal item, search plans, and similarity metrics, the RETRIEVER selects a target destination and search plan to execute once SHOPPER arrives. Table 1 lists conditions for selecting a particular destination and search plan.

Recall earlier that SHOPPER uses regularities of a domain in order to find items via its search plans. Regularities are also used for deciding on a destination. For the examples discussed later, we use two regularities: *type* and *counterpart*. The type relationship denotes a category made up of items of the same type. The counterpart relation denotes a category of items that are often used together; e.g., pancake mix and maple syrup.

As an example of how the type relationship is used, Surf and CheerFree are types of laundry detergents. Items of the same type are likely to be physically close. If, in a previous visit, SHOPPER found Surf and now wants to find CheerFree, it selects a place of interest (Surf) as the destination as well as a LOCAL SAME-SIDE search plan. This particular search plan looks for an item hypothesized to be nearby on the same side of the aisle as the previously found item.

Path Planning

Given a target destination from the RETRIEVER and the current location from the map, the PATH PLANNER plans a route that will get the agent from the current location to the destination. Because the map is organized in terms of nodes and edges, the path planner uses Dijkstra's algorithm for finding a shortest path. No metrical information is stored, so each edge is of equal cost. After the nodes along the route have been selected, the PATH PLANNER then annotates each node with all the sensing information gathered from past visits. These annotations are used by the FOLLOWER.

Route Following

The FOLLOWER receives a path and search plan from the PATH PLANNER. The FOLLOWER's purpose is to follow the path and then pass the search plan to the PLAN INTERPRETER. In order to follow a path the FOLLOWER must verify that the current sensing is consistent with the predicted sensing. The stored sensing is processed according to the particular place prediction. Recall these are a set of tuples $\langle T, S \times C, A, R \rangle$.

The consistency check is based on a match function for each sensing routine:

$$\forall s \epsilon S \exists m_s : A \times R \times A \times R \rightarrow \{True, False\}$$

where m_s is the match function for sensing routine s. The match functions compare the arguments and results of the past and current sensing routine to ensure the agent is on course. If the results are consistent, the match function returns a match (True), otherwise no match (False). For example, suppose SHOPPER is checking if shelf positions match. After the Shelf Detector is run on the current image, the arguments (horizontal subsampling) and currently found shelves (vertical image positions) are passed to the shelf match function as well as the stored shelf information composed of the same arguments and vertical image positions. For this particular match function, we require one-third of the shelf positions in the current image be within twenty pixels of the stored shelf positions, and vice versa. If both match correctly, the shelf match function returns True otherwise False.

The consistency check is done by using the match functions over the sensing routines common to both the past and current perception. Let P be the set of stored perception at a place, and let Q be the set of current perception. Place consistency is defined to be equivalent distinctive place types and a match on all the common sensing between P and Q. Figure 5 illustrates this method as a procedure. If the procedure returns True, the two locations are consistent. If False, the two locations are different.

```
procedure Consistency-Check(P, Q)
  for all sensors s ∈ S
    for all compass directions c ∈ C
      if ⟨t, (s, c), a, r⟩ ∈ P and
         ⟨t', (s, c), a', r'⟩ ∈ Q and
         m_s(a, r, a', r') = False then
         return False
  return t ≟ t'
```

Figure 5: Procedure for determining consistency between places P and Q.

This procedure is at the heart of the PASSIVE MAPPER. The FOLLOWER performs the stored sensing routines, and then makes use of the procedure to ensure it is on course. At an intersection the FOLLOWER checks for consistency by alternating between a consistency check and executing a sensing routine. After the intersection matches, the FOLLOWER orients the agent

Strategy Name	Conditions	Destination	Search Plan
EXACT	I_1 was found before.	POI with I_1	None
TYPE	Similar item I_2 was found before.	POI with I_2	LOCAL SAME-SIDE
COUNTERPART	I_1 and I_2 are counterparts. I_2 previously found.	POI with I_2	LOCAL
SIGN-TYPE	Sign S seen. I_1 is a type of S.	INTER with S	AISLE
SIGN-COUNTERPART	Sign S seen. I_1 and S are counterparts.	INTER with S	AISLE
DEFAULT	None	None	BASIC

Table 1: In order of preference: strategy names, their conditions, destination, and search plan for locating item I_1.

to move toward the next distinctive place. A similar method is employed for finding a place of interest, except that a consistency check failure is allowed owing to the agent not being in the right place yet. Currently SHOPPER does not handle the case when it fails to find a place of interest.

Examples

We select three items to demonstrate SHOPPER's navigation capabilities: Solo laundry detergent, Corn Pops cereal, and Downy fabric softener.

Solo Initially in first coming to the store, SHOPPER's map is empty. Given an empty map, Solo laundry detergent, and preferences shown in Table 1, the RETRIEVER picks a null destination and BASIC search plan as the DEFAULT strategy. The BASIC search plan is simple: go to the beginning of an aisle, move across aisles until a relevant sign is seen, go into that aisle, look left and right until the item is found.

On receiving the null destination with search plan, the PATH PLANNER outputs a null path and BASIC search plan. The FOLLOWER has no path to follow, so it passes control and the search plan to the PLAN INTERPRETERstarts executing the search plan starting at the store entrance in front of Aisle 1. The BASIC search plan instructs SHOPPER to move around the outside perimeter of the store reading signs and moving until it finds a relevant sign. For example, in Aisle 4, the sign reads: Aisle-4 Salad-dressing Canned-soup Sauce Nut Cereal Jam Jelly Candy. Eventually Solo is found on the left side of Aisle 6 since there is a "laundry aid" sign in front of that aisle. During the time this search was done, the PASSIVE MAPPER recorded the intersection types of Aisles 1 through 6 plus visual (sign) information. A place of interest is created where Solo was found. The POI is defined according to the shelf positions, color region, and item identification as output by the sensing routines.

Corn Pops Next, we give SHOPPER the goal of finding Corn Pops. The RETRIEVER recalls that a "cereal" sign was seen in front of Aisle 4 and selects the SIGN-TYPE strategy. The target destination is now the beginning of Aisle 4, and the search plan is AISLE. The PATH PLANNER plans a path from the current location (a POI) to Aisle 4 (an INTER). The FOLLOWER starts by verifying it's at the current place. The current accumulated sensor information is a match to the stored perception. SHOPPER now orients its body to the beginning of Aisle 6 and goes there. Once it reaches the beginning, the intersection is verified to be Aisle 6 by matching the intersection type and turning the head around to match sign information. Using a similar method, SHOPPER continues to the beginnings of Aisle 5, then Aisle 4. The AISLE search plan and control is now passed to the PLAN INTERPRETER. The PLAN INTERPRETER searches Aisle 4 and eventually finds Corn Pops where it creates a POI similar to Solo.

Downy Finally, we give SHOPPER the goal of finding Downy fabric softener. Since fabric softener is used with laundry detergent, SHOPPER uses the COUNTERPART strategy. The intended destination now becomes the POI containing Solo, with the search plan as LOCAL. In a similar fashion, the FOLLOWER follows a route from the current POI to the intended POI. When the agent is at the beginning of Aisle 6, the FOLLOWER then runs the POI sensing routines and compares results associated with Solo while moving down the aisle. Once Solo is reached, a LOCAL search plan is executed. This plan allows SHOPPER to search on the left and right of Solo as well as the other side of the aisle as tries to find Downy. In this instance, Downy is on the other side. SHOPPER finds Downy and creates a new POI.

Status

SHOPPER has been tested successfully on fourteen items ranging over cereals, laundry aids, cake mixes, cleaning materials, and storage supplies. SHOPPER can quickly compute routes to likely areas and reliably arrive there. For the items SHOPPER cannot find – and there are many – it has been the case that its sensors failed to detect the item. So long as SHOPPER reaches a close enough proximity to point the camera at an item, we do not consider the navigation system faulty.

Currently the FOLLOWER uses a static procedure for following routes. Because our search plans are declarative and can account for opportunistic types of behavior (e.g., recognizing a sought item unexpectedly), we would like the FOLLOWER to use a similar representation for coping with contingencies during the navigation process, c.f. (Simmons 1994).

Earlier we cited the possibility of the system using different or new sensing routines, not necessarily having any overlap with previously stored sensing. We believe that landmark disambiguation is simpler if the PASSIVE MAPPER is sometimes active by signaling an ambiguity before the agent moves away. Then it duplicates perceptual actions and compares the results to past perception. This method appears to be promising since Kortenkamp and Weymouth, in using a visual representation of vertical lines, were able to successfully disambiguate locations without traveling away from the location.

Another possible way to disambiguate position is to use dead reckoning. Currently SHOPPER's map data does not indicate relative distances between places. So when sensor data alone indicates that SHOPPER could be in one of several places, dead reckoning allows SHOPPER to reject many places before needing more information.

The navigation method we have presented here assumes that major errors in sensing will not happen. For a specific set of items, our sensing routines have empirically shown to be sufficient and reliable in our particular domain. For integration with existing robots, this may not be a realistic assumption. However, we view errors in sensing as being just that: errors in sensing. We do not believe a mapper should bear the burden of coping with an incorrect map because of error-prone and/or semantic-poor data. Surely there are instances in real life where one can become genuinely lost because of sheer size, or absence of distinguishable cues. Although every navigation system must handle those inevitable situations, we believe those instances are rare simply because we live and depend on a culturally rich world (Agre and Horswill 1992) full of distinguishing cues to support everyday activity - one of them being navigation.

Acknowledgments

The research reported here has benefited from discussions with Charles Earl, Mark Fasciano, Jim Firby, and Val Kulyukin as well as from helpful comments from reviewers. This work was supported in part by Office of Naval Research grant number N00014-91-J-1185.

References

Philip E. Agre and Ian Horswill. Cultural support for improvisation. In *The Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 363–368, 1992.

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

Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, 1987.

Sean P. Engelson and Drew V. McDermott. Maps considered as adaptive planning resources. In *AAAI Fall Symposium on Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots*, pages 36–44, 1992.

R. James Firby, Roger E. Kahn, Peter N. Prokopowicz, and Michael J. Swain. An architecture for vision and action. In *The Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 72–79, 1995.

Daniel D. Fu, Kristian J. Hammond, and Michael J. Swain. Action and perception in man-made environments. In *The Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 464–469, 1995.

Ian Horswill. Polly: A vision-based artificial agent. In *The Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993.

Hiroshi Ishiguro, Takeshi Maeda, Takahiro Miyashita, and Saburo Tsuji. A strategy for acquiring an environmental model with panoramic sensing by a mobile robot. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 1, pages 724–729, 1994.

David Kortenkamp and Terry Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *The Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1994.

Benjamin J. Kuipers and Yung-Tai Byun. A robust, qualitative method for robot spatial learning. In *The Proceedings of the Seventh National Conference on Artificial Intelligence*, 1988.

Maja J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312, June 1992.

William Rucklidge. *Efficient Computation of the Minimum Hausdorff Distance for Visual Recognition.* PhD thesis, Cornell University Department of Computer Science, 1994. Technical Report TR94-1454.

Reid Simmons. Becoming increasingly reliable. In *Proceedings, The Second International Conference on Artificial Intelligence Planning Systems*, pages 152–157, 1994.

Thomas M. Strat and Martin A. Fischler. Context-based vision: Recognizing objects using both 2-d and 3-d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, 1991.

Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.