Detecting and Diagnosing Navigational Mistakes

Stuck, E.

Publisher's version / Version de l'éditeur:

Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '95), 1995

In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, August 5-9, 1995.

Detecting and Diagnosing Navigational Mistakes*

Elizabeth R. Stuck
Institute for Information Technology, National Research Council
Montreal Road, Ottawa, Ontario K1A 0R6
liz@ai.iit.nrc.ca

Abstract

The characteristics of natural environments, as well as the limitations on representations, sensors, and actuators, make navigational mistakes inevitable. This paper looks at how to detect and diagnose mistakes autonomous mobile robots make while navigating through large-scale space using vision. Mistakes are perceptual, cognitive, or motor events that divert one from the intended route. Detection and diagnosis consist of realizing a mistake has occurred, determining what it was, and when it happened.

This paper describes an approach that detects mistakes by finding mis-matches between observations and expectations. It diagnoses mistakes by examining knowledge from a variety of sources, including a history of observations and actions. It supports these operations by using symbolic visual information to compare expectations with observations augmented by a priori *knowledge. This paper describes* MUCKLE*, the simulation used to test the approach, and presents experimental results that demonstrate its effectiveness.*

1 Introduction

Navigation is problematic. Sensors cannot measure the environment precisely or completely. The information they provide contains quantitative and qualitative error. Models of this information are limited by the representational formalisms used. The limitations of actuators also result in error that ranges from inaccuracy to gross failure. In addition, natural environments are constantly changing. Objects move and change appearance. For these reasons, failures in navigation are inevitable. Autonomous robots must handle them to navigate successfully.

The research described in this paper distinguishes between mistakes and the more common concept of error. For the most part, the concept of error in mobile robotics research encompasses only numeric variation, which is not sufficient to model mobile robot behavior. Occasionally, quantitative variations are so large that results are qualitatively different than expected. For example, a robot may turn too late onto the wrong path because its odometry is significantly off. In addition, much information that is relevant in navigation is symbolic and qualitative, such as an object's identity and composition. In such cases, the idea of quantitative error no longer applies. This paper uses the term "mistake" to refer to both kinds of categorical inaccuracies, numeric and symbolic.

Because much of mobile robotics research defines error quantitatively, most approaches deal only with quantitative navigational failures. Those few that do handle qualitative failures deal for the most part solely with unexpected obstacles. In addition, only a few navigational systems use visual landmarks, and they do not deal explicitly with incorrectly identifying landmarks. In general, most navigational systems have developed strictly local strategies for dealing with navigational failures.

This paper discusses an approach for detecting and diagnosing mistakes that autonomous mobile robots make while navigating through large-scale space using vision [8]. A mistake is a perceptual, motor, or cognitive event which causes one to stray significantly from an intended route. Detecting a mistake consists of inferring from one or more symptoms that a mistake has occurred. Diagnosing a mistake involves investigating the symptoms to find their cause and determine what the mistake was as well as when or where it occurred.

Detecting and diagnosing mistakes entails searching through a large problem space of actions and perceptions to ascertain which of them may be incorrect. An important characteristic of this problem is that mistakes usually do not manifest themselves immediately. They are often discovered a significant amount of time and/or distance after they occur, through apparently unrelated symptoms. As a result, the space of possible mistakes is too large to search without using heuristics to prune it.

Detecting and diagnosing mistakes requires all of the available relevant information to guide and constrain search. A solution to this problem consists of four parts. Visual information is represented using a detailed symbolic formalism to capture the information essential for navigating and detecting and diagnosing mistakes. This information can then be used to compare observations and expectations augmented by *a priori* knowledge. Mistakes are detected by identifying mis-matches between these observations and expectations. Finally, mistakes are diagnosed by examining knowledge from a variety of sources, including a history of observations and actions.

This approach has been implemented and tested using MUCKLE, a system that simulates visual and motor capabilities. Using a simulation has enabled the research to focus on developing formalisms and strategies for detection and diagnosis without having to implement general vision and motor control systems. Simulations involve pitfalls, however; the simulation avoids them in part in two ways. One aspect of navigation is that it requires filtering an immense amount of data and selecting what is relevant. The simulation incorporates this by including irrelevant data. Another characteristic of robots is that their capabilities for sensing and motor control are imperfect. The simulation models this as well.

2 Related work

Many researchers in mobile robotics have developed reactive approaches to navigation [3]. However, they do not, in general, deal explicitly with mistakes. Such systems do not have explicit expectations, and so for them, there are no unexpected situations. Having expectations requires pulling together a variety of information that is typically distributed or not present at all in a purely reactive system. In addition, reactive systems may have a destination, but often no subgoals or specified route to follow. As a result, it is not possible to *not* follow the route correctly. Running into obstacles or interpreting sensor data incorrectly are the only kinds of mistakes possible.

Traditional approaches to mobile robot navigation are based on motion planning (e.g., [9]). Many of them detect only a limited kind of mistake: unexpected obstacles that prevent a robot from following a path. Many of these systems handle mistakes simply through prevention, which usually consists of redundant low-level image processing techniques. A somewhat more sophisticated system is Hilare, which can also detect absent or incorrectly located beacons and failing sensors [4].

A recent trend in mobile robotics is to combine reactive and deliberative navigation to produce fast, robust, intelligent behavior. Most of these approaches de-emphasize the perceptual aspects of the problem, and do not model important and problematic characteristics of visual information. The approach developed by Noreils, for example, handles external faults, including changes in the environment, and internal faults involving robot hardware and sensors, but does not deal with visual mistakes [7].

Another approach used is model-based navigation. FINALE uses a model of the robot's position uncertainties to guide scene interpretation [5]. It treats unexpected objects as obstacles, and if it can't match its expectations and perceptions, it turns and tries again. This, like many model-based approaches, requires a geometric model of the environment, which can be expensive to generate.

Landmark-based navigation relies on *a priori* knowledge of landmarks to guide motion (e.g., [2,6]). Much of this work deals more extensively with visual and motor mistakes. However, most of the mistakes are quantitative, not qualitative. One exception is ELMER, which detects certain qualitative errors by adding explicit "error transitions" to plans and monitoring plan execution [10]. Errors are found by comparing events (features seen or distances travelled) that occur during current and past plan executions. However, this work lacks a thorough treatment of mistakes and mechanisms for detecting them, and does not address diagnosis, which can aid and inform recovery.

3 Solution

This section discusses the four components of the approach: formalisms for representing symbolic visual information, methods for comparing this information, and heuristics for detecting and diagnosing mistakes.

3.1 Representing symbolic visual information

Detecting and diagnosing mistakes relies on representing detailed information about expectations and perceptions about the environment. The components of the environment, such as things, places, paths, and portals, are called *entities*. Entities have three types of attributes that play a vital role in navigation. Attributes for recognition include position-independent information such as the entity's identity, size, and shape. Attributes aiding in localization include the direction the entity is facing and its direction and distance from the viewer. Attributes which are useful in detecting and diagnosing mistakes include what other entities this entity can be confused with, as well as how likely it is to move from its location.

The navigation system uses a route description, an ordered list of one or more instructions that describe how to reach a destination. *Prescriptive* instructions specify motions to execute in terms of distances, directions, and entities. *Descriptive* instructions specify entities the robot should and should not see while navigating.

3.2 Comparing symbolic visual information

A crucial component of mistake detection and diagnosis is comparing visual information. First, a robot must compare its expectations with its perceptions to determine if it is where it expects to be. Second, a robot must be able to identify entities that do not match its expectations, in case this information is needed later to diagnose mistakes. Since qualitative results do not provide enough information to make navigational decisions and detect and diagnose mistakes, the comparison process produces a real-valued number between 0.0 and 1.0, inclusive, called *compatibility*, which describes the quality of a match between groups of entities as well as single entities.

The first task is formulated as finding the best correspondence between a set of expected entities generated from the route instructions and a set of perceived entities. The best correspondence has the highest average compatibility of the component entity matches. Finding the best correspondence requires comparing expected and perceived entities and calculating the compatibility for each pair. To allow for no correspondence being found for an expected entity, the list of perceived entities includes a "null entity" that can match any expected entity. (This is similar to the idea of nilmapping [5].)

The second task is formulated as finding the best match for each perceived entity from the set of generic entities, which have been provided *a priori*. The best match is the match with the highest compatibility. Finding the best match requires comparing perceived and generic entities and calculating their compatibilities.

Finding the best correspondence in both cases is problematic because it is not computationally feasible to compute all possible correspondences when these sets have more than five or six elements. This process is constrained using a generic entity taxonomy based on similar attributes. Only very similar entities are compared at first; if no matches are found, this constraint is relaxed.

Calculating the compatibility of two entities requires finding the similarities between their attributes and combining these values. This is done using the Dempster-Shafer theory of evidential reasoning as reformulated by Andress and Kak [1]. This theory takes as input three belief measures about each attribute value match: similarity, dissimilarity, and ignorance. MUCKLE calculates these belief measures using look-up tables for symbolic values and arithmetic functions for numeric values.

3.3 Detecting mistakes

To detect mistakes, the robot generates expectations, compares these expectations with its observations, and determines when they do not match. *Visual* expectations provide primarily local guidance for executing individual instructions. Information sources for visual expectations include instructions and *a priori* generic knowledge. For example, knowledge about an entity's usual size may be used to fill gaps in information provided by an instruction. *Motion* expectations from the route description provide local and global guidance. Local guidance tells how far the robot should travel to execute an instruction. Global guidance tells how far the robot should travel to execute the route description and in what direction the goal lies.

The robot generates rough local and global expectations in advance and refines them while navigating. Initially, the robot uses the distances specified in instructions to calculate the *mean instruction duration* (MID), an estimate of the average distance that each motion instruction covers. The robot uses this estimate to add quantitative information to instructions that express distances qualitatively, in terms of landmarks. Then it calculates the direction and total distance to the destination, using the distance and direction specified by each prescriptive instruction. As the robot executes instructions, it updates estimates of the direction to the destination, the average distance of instructions, and how far it has travelled since it began executing the current instruction and the route description. It updates its estimates using information based on the actual results of executing instructions so far and the unexecuted remaining instructions. This information allows the mobile robot to decide if an instruction or the entire route is taking too long to complete.

One set of perceptions and one instruction alone do not contain enough information for the robot to decide whether it has made a mistake. It is necessary to combine the results over time. This process is made more difficult because the expectations used for comparison involve disparate information: distances, directions, and entities that should be seen at the beginning or end of an instruction, sometime, never, or continually.

The solution to this problem has three parts. A real-valued number, called *conviction*, represents the mobile robot's certitude that it has not made any mistakes. To incorporate the uncertainty about distance travelled, the conviction decreases by a small amount after each of the robot's motions. *Sigmoid* functions then combine various *consistency* values, which are the results of the comparisons, into a single new measure of conviction. These functions take two inputs, the current conviction and a consistency measure, and produce one output, a new conviction. These functions are called sigmoid because they are the three-dimensional analog of an s-shaped curve.

3.4 Diagnosing mistakes

This approach deals with three classes of mistakes.

False negatives occur when the robot fails to see an entity. They depend on a variety of factors, including lighting, unusual viewpoints, and attention. These mistakes usually lead to travelling too far. *Mis-recognition* mistakes occur most frequently with entities that look similar. These mistakes include seeing an entity and either incorrectly identifying or failing to identify it as an expected landmark. The former type of mistake leads to travelling too far, and the latter to not travelling far enough. *Motor control* mistakes consist of problems with steering and locomotion. Errors in estimating motion also fall into this category. Consequences of these mistakes include travelling too far, not far enough, or turning incorrectly.

Diagnosing a mistake consists of determining what caused the robot to stray from its route, and when or where it happened. A specification of the "what" includes the events making up the mistake, the general type of mistake, and the consequence of the mistake. A specification of the "when" includes the instruction the mistake occurred during and the distance or time after starting to execute the instruction that the mobile robot made the mistake. To provide the necessary information to diagnose mistakes, the mobile robot keeps a record of all of its motions, perceptions, and comparisons while navigating. Also useful are certain kinds of *a priori* knowledge, such as what entities are confusable with other entities.

The diagnostic process has four steps. The first step is to analyze and *calibrate* the history. This produces several statistics, including distances of the instructions that took the least and most distance to complete, the MID, and its standard deviation. Another statistic is the mean distance interval, which begins at the shortest distance and ends at the MID plus the standard deviation.

The second step consists of applying three classes of strategies to *generate* evidence about possible mistakes. One class searches for evidence of mis-recognitions that indicate the robot should have completed an instruction later than it actually did. Another class searches for evidence of mis-recognitions or false negatives that suggest the robot should have completed an instruction sooner than it did. A third class of strategies looks for evidence of motor control mistakes, by searching for alternate paths that lie in directions similar to the robot's motion.

These strategies generate many pieces of evidence about possible mistakes. The third step of the diagnostic process uses a set of heuristics to *refine* this evidence and generate hypotheses about what mistake occurred and when. These heuristics uses the results of visual comparisons to rank a hypothesis based on whether it relies on the best match at that location or one or several good matches.

The fourth step uses heuristics about the timing of the hypothesized mistake to *select* the best hypothesis. One heuristic eliminates evidence that suggests a mistake has occurred very early or very late in executing an instruction, since a mistake is unlikely to occur then. Another heuristic eliminates hypotheses inconsistent with the instruction's duration. For example, a hypothesis is inconsistent if it asserts the robot stopped too soon while executing an instruction and if the instruction's duration is longer than the MID. A third heuristic favors hypotheses about mistakes that occur near the MID over those that occur much earlier or later. The fourth heuristic, which uses the mean distance interval, is less reliable and is used only to resolve ties. This heuristic relies on the tendency for instructions in a particular route description to take the same order of magnitude of distance to execute.

4 Implementation

MUCKLE, the system implementing this approach, consists of two parts: the navigation and simulation systems. The navigation system interprets route instructions that have been generated by hand and produces motion commands. The simulation system includes a representation of the environment, a vision simulator, and a motor control simulator. This section discusses the simulation system and presents the experimental results.

4.1 Simulation system

The *environmental map* represents the world that the simulated autonomous mobile robot moves in. It consists of thirty-six types of entities, including various kinds of paths, places, portals, and things. It includes several kinds of entities not useful in navigation to make the problem more realistic, such as bushes, light poles, and people. These entities may interfere by occluding other significant entities, by adding to the possibilities for identity confusion, or by blocking the robot's path unexpectedly. The map consists of a two-dimensional array. Each cell represents a one meter square area. (In the examples discussed in this section, the environment is represented by an array of 50 by 50 cells.) Each cell contains two pointers to the static entity and dynamic entity, if any, occupying the cell.

The *vision simulator* uses the environmental map to construct an iconic view, which is a two-dimensional perspective view of the environment from the simulated robot's perspective. The iconic view consists of a two-dimensional array in which rows correspond to altitude and columns to direction. Cells contain a pointer to the entity seen at that altitude and direction plus its distance. The vision simulator incorporates visual error by corrupting position-independent entity attributes and distorting the environmental map before creating the iconic view.

The *motor control simulator* simulates the execution of motion commands. At any one time, the simulated robot occupies only one cell and has one of eight possible orientations, which are multiples of 45 degrees. The simulated robot can move from the cell it is occupying to any one of that cell's eight neighbors in one time interval. As a result of this coarse discretization, only certain types of motion mistakes are possible. The size of the error that the simulated robot may make in turning can only be a multiple of 45 degrees. This error will occur only when there is another path present in the incorrect direction. Another type of mistake is that the simulated robot may either underestimate or overestimate its motion.

4.2 Experimental results

Eight experiments were run using MUCKLE. MUCKLE correctly detected that a mistake occurred in all eight experiments. It correctly diagnosed the mistake in five of them (A, C, D1, D2, and E2). One of these experiments, A, is described in detail below. Table 1 summarizes the results of the remaining experiments.

Figure 1 shows the environment of experiment A and the simulated robot's path. The simulated robot and its path are represented by black shapes resembling a Pac-man. The environment is populated with a bush, mailbox, signpost, signboard, flagpole, statue, bench, set of stairs, and foot-bridge; five buildings, four hedges, three bicycle racks, four trees, and two roads; and many light poles, sidewalks, grass plots, and sidewalk intersections.

Table 2 shows the route description the simulated robot used to navigate in experiment A. Figure 1 shows the correct route it took in the first trial when it did not make any mistakes. In the second trial, the environment also contained a person, in the sidewalk intersection to the left of the flagpole. Figure 2 shows the path the simulated robot took when it incorrectly identified the person as the statue mentioned in the sixth instruction. As a result, the simulated robot stops executing instruction (6) too soon.

In this experiment, the simulated robot detects that it has made a mistake and stops when it reaches the road. Its conviction drops below threshold here, since it expects to be travelling on a sidewalk. After generating hypotheses, there are two hypotheses that are the most likely. The first is that while executing instruction (4), the simulated robot should have seen a flagpole and stopped sooner, when it was at the first sidewalk intersection after turning left at the bench. The second hypothesis is that the simulated robot mis-recognized the person as a statue while executing instruction (6). The simulated robot correctly chooses the second hypothesis because the duration of instruction (6) is closer to that of the shortest instruction than to the mean instruction duration (MID), making this instruction more likely to have been incorrectly executed.

(1)	begin on a sidewalk
(2)	go to sidewalk intersection with bench to left
(3)	turn right 90 degrees
(4)	go to sidewalk intersection, flagpole to right
(5)	turn right 90 degrees
(6)	go to sidewalk intersection near statue
(7)	turn left 90 degrees
(8)	go to foot-bridge
(9)	stop on foot-bridge

Table 2: Route description (experiment A).

5 Discussion

In all eight of the experiments, MUCKLE correctly detected a mistake. In five of the eight, it correctly diagnosed the actual mistake. In the rest, it twice ranked the actual mistake second and once ranked the actual mistake third. This performance is reasonable given the nature of the problem. On average, over 200 pieces of evidence are generated in each experiment. The diagnostic process is able to refine this large amount of data into a small number of hypotheses and ranks the actual hypothesis as no worse than third in all of the experiments.

The experiments run test many aspects of this

	Mistake				Consequence			Symptom			Diagnosis
	-visual-		-motor-		too	too	wrong	low	too	dead	actual
Experiment	mis-recognition	false negative	distance	direction	soon	late	way	conviction	far	end	mistake
A	•				•			•			1st
B1		•				•		•			2nd
B2	•					•		•			3rd
C		•				•			•		1st
D1			•		•			•			1st
D2			•			•				•	1st
E1				•			•	•			2nd
E2				•			•			•	1st

Table 1: Experimental results.

approach for detecting and diagnosing mistakes. The experiments provide good coverage of the many navigation mistakes possible. The results also demonstrate the usefulness and feasibility of this approach. The results do suggest that one of the heuristics, however, is not reliable enough. The heuristic that uses the mean distance interval is responsible for all three diagnostic errors, and only aids in one correct diagnosis. The performance of the system could be improved by refining or replacing this heuristic.

This approach will not always correctly diagnose the mistakes it has detected or even detect all of the mistakes that have occurred. In fact, it will occasionally detect phantom mistakes. However, this should not be surprising, since one of the difficulties inherent in navigation and thus in detecting and diagnosing mistakes is that the information available is inaccurate and incomplete.

An important limitation of the research described in this paper is using a simulation for testing. This allowed the research to focus on the problem of reasoning about navigational mistakes, but also made it possible to simplify many aspects of mobile robotics, including characteristics of motor behavior and visual information. A clear direction for future research is to modify this approach and implement and test it on a mobile robot. One way to provide a limited form of the required object recognition would be by labelling objects with bar codes.

Acknowledgements

I thank Professor William Thompson, who supervised this research and provided valuable insights into this problem. I also thank anonymous reviewers for their constructive comments on earlier versions of this paper. This work was supported by National Science Foundation grant IRI-9196146, with partial funding from the Defense Advanced Research Projects Agency.

References

[1] K. M. Andress and A. Kak. The PSEIKI Report-Version 3: Evidence accumulation and flow of control in a hierarchical spatial reasoning system. Robot Vision Lab TR-EE 89-35, Purdue Univ., West Lafayette, IN, Nov. 1989.

[2] R. C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105-122, 1990.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. of Rob. Auto.*, RA-2(1):14-23, March 1986.

[4] M. Ghallab. Task execution monitoring by compiled production rules in an advanced multi-sensor robot. H. Hanafusa and H. Inoue, editors, *Robotics Research, the Second Int. Symp.*, pp. 393-401, Cambridge, MA, 1985.

[5] A. Kosaka and A. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: IU*, 56(3):271-329, Nov. 1992.

[6] T. S. Levitt, D. T. Lawton, D. M. Chelberg, P. C. Nelson. Qualitative landmark-based path planning and following. *Sixth Natl. Conf. AI*, pp. 689-694, Seattle, WA, July 1987.

[7] F. Noreils. Integrating error recovery in a mobile robot control system. In *IEEE Int. Conf. on Robotics and Automation*, pp. 396-401, Cincinnati, OH, May 1990.

[8] E. R. Stuck. *Detecting and diagnosing mistakes in inexact vision-based navigation.* PhD thesis, Dept. of Comp. Sci., University of Minnesota, Minneapolis, MN, Nov. 1992.

[9] C. Thorpe, M. Hebert, T. Kanade, S. Shafer, and the members of the Strategic Computing Vision Lab. Vision and navigation for the Carnegie-Mellon Navlab. *Annual Review of Computer Science*, 2:521-556, 1987.

[10] B. Ward and G. McCalla. Error detection and recovery in a dynamic planning environment. In *Second National Conf. on AI*, pp.172-175, Pittsburgh, PA, Aug. 1982.

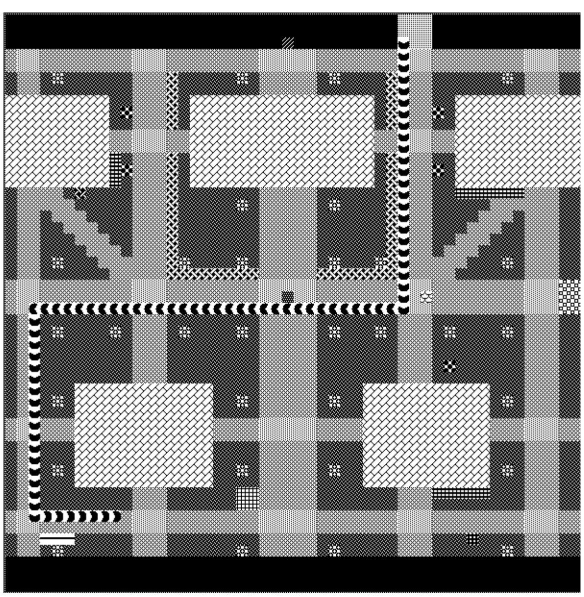

Figure 1: Correct route (experiment A).

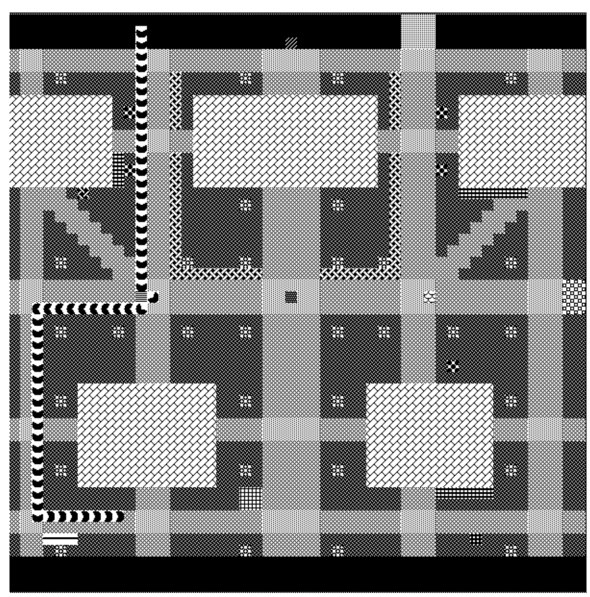

Figure 2: Incorrect route (experiment A).