EXPERIMENTS IN THE INTEGRATION OF WORLD KNOWLEDGE WITH SENSORY INFORMATION FOR MOBILE ROBOTS*

A. C. Kak, B. A. Roberts, K. M. Andress and R. L. Cromwell

Robot Vision Lab
School of Electrical Engineering
Purdue University
W. Lafayette, IN 47907

ABSTRACT

World representation is obviously a most important part of the sensory intelligence required by a mobile robot in order to operate autonomously in a complex environment. In this paper, we will propose a taxonomy of robot intelligence based on the methods used for world representation and for the integration of sensory information with the representations. These methods of representation and sensory integration are currently being programmed into PETER, a mobile robot, in our lab.

1. INTRODUCTION

Our aims in mobile robot research are rather modest, although still very challenging. Unlike many other studies in the past, we do not want our robot to operate in an unstructured and unknown environment. We do not wish to provide our robot with map making capability with the use of sonar or lasers.

On the other hand, the robot in our research is provided a road map -- a sidewalk map, to be precise -- with all the intersections and turn points clearly indicated, and the aim of the robot is to use its sensors to travel between two specified points on the map. One could say that our emphasis is more on how to process sensory information reliably so as to generate confirmatory evidence for the information represented in the map.

To impress upon the reader the fact that our goals, although less general than that of a robot in an unknown environment, are still very difficult to attain, consider the following two complicating factors, which by no means possess immediate solutions:

- To keep the robot reasonably autonomous, we have intentionally chosen not to use beacons that could be used for obtaining position fixes. The robot must use encoders on its drive and steering motors to keep track of its position and orientation. Unfortunately, because of slippage in the wheels and other factors, this technique for position and orientation determination is error prone, and, more important, the uncertainty box increases in size as the robot gets farther away from the start position. The uncertainty box is the region made up of all possible robot locations which could result from the position uncertainties. *Fortunately, it is possible to zero out the uncertainty box at those turn-points and intersections that can be correlated with large confidence factors with information on the map.*
- The photometric sensors for generating sidewalk edge information are bound to be unreliable. As a result, the sidewalk definition obtained from sensors will be discontinuous and there will be false indications of sidewalks where none exist. When these errors in perceived information are in an uncertainty box that includes a turn point, it is entirely possible that the robot will make an incorrect decision and will either get off the sidewalk or turn on to the wrong sidewalk. The important question then becomes: *What metrics does the robot use to ascertain that it has made an incorrect decision. Of course, considerations such as a diverging direction of travel with respect to map indications and increasing distance away from the next turn point, should enter the design of an appropriate metric.*

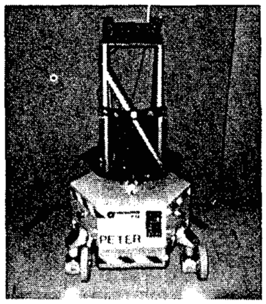
In this paper, we will address these and other related issues. We will also present a taxonomy for classifying the intelligence of the kinds of robots we are interested in. Two of the three modes of world representation and sensory integration that form the taxonomy are at this time being programmed into PETER, a mobile robot, in the Robot Vision Lab at Purdue. A picture of PETER is shown in Fig. 1.

In order to place our work in a proper context, we will now present a brief survey of previous work in autonomous mobile robots:

Elfes [1] has described a sonar-based mapping and navigation system for robots in an unstructured and unknown environment. Sonar range data is used to build maps for navigation of an autonomous robot through an unknown environment. The maps make up a multi-leveled description of the environment by describing regions as *occupied, empty,* or *unknown.* This method requires the combination of information from many different sensor positions. Several sonar sensors are used, and the robot can, of course, be reoriented to allow for more sensor positions at any given location.

Brooks [2] discussed a layered control system, used for the control of a wandering mobile robot. The task of controlling the behavior of the robot is split into a hierarchy of tasks that are based on behavioral rather than functional actions. The low-level tasks include avoiding obstacles and wandering about. Higher-level tasks include building maps of the regions traversed, and reasoning about the objects in the robot's environment. This would allow a robot to start with no knowledge of its environment, and build up a map as it wanders about, learning more about the environment and the objects in it. The various layers (or modules), operating in parallel, contrast with traditional pipeline architectures for mobile robot control.

*Work supported by National Science Foundation.



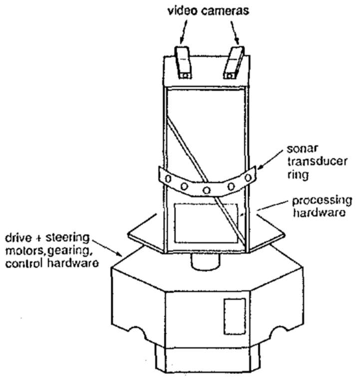


Fig. 1: On the left is a photo of PETER and on the right a description of various features on the robot.

Harmon [3] has described a blackboard-based path planning system for an autonomous robot with no *a priori* knowledge of the terrain it is to navigate. Multiple sensors gather information on the vehicle's immediate surroundings. This information is combined with knowledge of human orienteering methods as well as statistics and information on terrain variability and obstacles in the database. Heuristics guide the search for a path through the environment to a goal destination.

Tsumura [4] has briefly mentioned a technique for maintaining the current position of an autonomous robot and determining the positional error. Some of these techniques might be useful if we wanted to provide a beacon at certain locations in the robot's world to allow it to correct its current estimate of position or orientation. Takeda et al. [5] have described a marker system to enable a robot to travel along a predetermined path.

The authors of [6] have described the control system of an autonomous vehicle that simultaneously plans its path, avoids obstacles, and follows a road. This system has a modular architecture that distributes the tasks among modules with anthropomorphic role definitions. Interaction takes place between modules performing in parallel. The design is an attempt to emulate the actions of a human team controlling the vehicle, a team including a pilot, a planner, an observer, a cartographer, etc.

Moravec [7] describes work with an autonomous cart equipped with video cameras. The cart employs stereopsis to determine the 3-D location of obstacles in its environment. A destination referenced to its starting location is specified and vision data is used to build a world model, which is in turn used by a path planning algorithm. After each one meter segment of the path is traveled, more vision data is collected and processed. After each short hop the vision data is used to verify the previous motion as well as for the planning of the remaining path. Although not all processing was done on-board, navigation was autonomous. The cart required several hours to navigate through a 20m course.

Crowley [8] describes navigation with the use of a dynamically maintained/updated local world model which is constructed through the integration of an a-priori global map and local sensor readings. The global map is a network of pre-learned places which are interconnected by legal highways. The composite local world model is constructed from sensor readings over a period of time from many views and from all sensors. It is this composite local world model that is used for local path planning.

Rao, Iyenger, Jorgensen and Weisbin [9] have presented concurrent algorithms for autonomous robot navigation in an unexplored terrain. They have proposed efficient data structures for the purpose of reducing navigation time and incorporating learning.

2. A TAXONOMY FOR MOBILE ROBOT INTELLIGENCE

Since world representation and sensory confirmation of that representation are central to the intelligence of an autonomous mobile robot, we believe that the level of intelligence of such a robot should be determined by these twin considerations.

The following modes of world representation and sensory integration, which form a taxonomy for mobile robot intelligence, are being programmed into PETER in our lab. The first two appear attainable in the next few years. As far as the third mode is concerned, we hope it proves to be better than a distant dream.

1. BAWR

   At the lowest level of intelligence, the robot uses a representation called the Blind Automaton World Representation (BAWR). In this approach, which is suitable for a sidewalk following robot, the network of sidewalks is represented by an attributed graph with a distinct arc for each straight segment; the attributes for arcs being the length of the sidewalk segment and its direction.

   Sensory integration for such a robot consists of using the map information to make the best guess from various alternatives that might be present in an uncertainty region surrounding an expected turn-point or an intersection. Other aspects of sensory feedback include evaluation of metrics for determining the appropriateness of the present position and orientation with regard to where the robot thinks it is on the map.

   Similar to navigation carried out by a blind person, the robot in this mode only knows that it must travel a certain distance in a certain direction to reach the next node in its itinerary. In drawing this analogy with a blind walker, we are assuming that in many cases a blind person is able to intuitively detect wrong paths selected, probably using factors such as greater than expected distance traveled in a particular direction, or by other clues regarding the nature of the sidewalk.

   In programming the robot in this mode, we are using a downward pointing camera to detect sidewalk edges in the immediate vicinity of the robot. Although a camera is used, we still call it a blind mode because the information generated by the camera is very similar to what blind walker might infer by using a stick to feel out the pavement.

2. NSAWR

   Near-Sighted Automaton World Representation (NSAWR) corresponds to the next higher level of robot intelligence. In this mode, we use higher level attributes for characterizing the arcs in the graph data structure for a network of sidewalks. These attributes are symbols corresponding to principal landmarks on either side, but only in the vicinity, of the sidewalk.

By using side-looking cameras, the robot tries to extract these symbols from the imagery by using large (fuzzy) operators -- the size of the operator depending upon how much data reduction is required before feeding the information into an inference processor. Corresponding to each operator size, every landmark is assigned a belief function value. Since for a large sized operator, it may be possible to confuse a utility pole with a tall parking sign, it is necessary that such ambiguities be accommodated by the assignment of operator-size dependent beliefs to the sensory outputs corresponding to the objects.

3. FVAWR
   The highest level of intelligence in the current scheme, which we believe is beyond the realm of possibility for the foreseeable future, will be achieved by a mobile robot employing Full Vision Automaton World Representation (FVAWR). For such a robot, the world representation would consist of a 3D model of the world it is supposed to operate in. By using active and passive sensors, such a robot would at each instant be aware of its environment, and would utilize this awareness for the purpose of navigation.

3. SENSOR MODELING

The robot must be able to cope with the following sources of error:

1. Errors in the position of the robot as derived from the drive motor encoders. The primary sources of this error are wheel slippage on loose surfaces and the quantization error of the shaft encoders that results from the play in the differential gears.
2. Errors in the orientation of the robot. The sources of these errors are the same as above, the encoder now being on the steering motor.
3. Errors in the detection of sidewalk edges, which lead to errors in the estimation of sidewalk centerlines. Sidewalk edges are detected by analyzing 2-D images of a downward-pointing camera. These errors fall into two categories: those that lead to breaks in the sidewalk centerline and those that generate false sidewalk edges and, therefore, false sidewalk centerlines. Note that the primary sensory modality for generating "world" information is photometric; ultrasonic sensors are also used, but only for bringing the robot to a halt if the possibility of an immediate collision is detected.

3.1 Position Errors

The position error is modeled by an uncertainty interval that increases in size as the robot gets farther away from the starting position. This is based on the assumption that error per unit distance traveled can be considered to be a constant, equal to 5% in our present programs. Figure 2a shows a linearly increasing interval along the centerline of a straight sidewalk segment. This figure also depicts the fact that in some cases it might be possible for the robot to correlate the turn-point that it has just traveled through with a high degree of confidence with the map information; when that happens, the robot can zero out the positional uncertainty box at the turn-point. If it is not possible to zero out the uncertainty at a turn-point or an intersection, the uncertainty interval keeps on increasing past that turn-point or intersection, as shown in Fig. 2b.

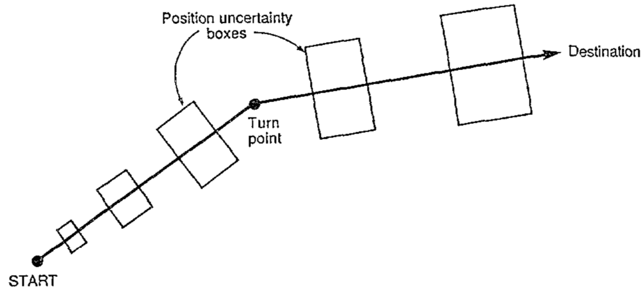


Fig. 2a: This figure illustrates a linearly increasing uncertainty box as the robot moves away from the starting position. If the robot can correlate the turn point with a high degree of confidence with map information, the uncertainty box is zeroed out at the turn-point.

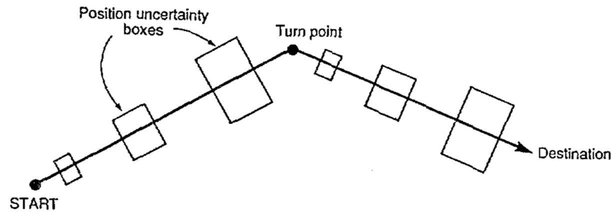


Fig. 2b: If the robot can not correlate with a high degree of confidence the turn-point with the map information, the uncertainty box continues to increase in size past the turn-point.

3.2 Orientation Errors

Steering commands are given to the robot based on the orientation of the chosen sidewalk centerline segment for the next 1 meter in front of the robot; of course, if a turn-point or an intersection is selected before the end of that distance, then a new steering command may have to be given at that point. We assume an initial uncertainty cone of $\pm 10^{\circ}$, as shown in Fig. 3a, and an increase in the size of this cone by one degree for every meter traveled. Figure 3b shows the increase in the uncertainty cone as the robot travels along a straight segment.

3.3 Sidewalk Detection Errors

Interpretation of vision data is carried out by using PSEIKI [10], a Production System Environment for Integrating Knowledge with Images. Figure 4 shows some sample images of the type seen by the downward pointing camera in BAWR mode of navigation; as mentioned before, these images only show a short stretch of the sidewalk immediately in front of the robot. The images on the right, used for drawing inferences about the location and the direction of the sidewalk, were obtained by applying edge detection and thinning operators to the images on the left. The architecture for PSEIKI is shown in Fig. 5. The process has two subsections: the preprocessor/pixel-to-symbol converter, and a rule-based edge labeler.

The preprocessor accepts digitized images from the robot and outputs binary edges suitable for conversion into symbolic form. The edges are detected by applying a Sobel

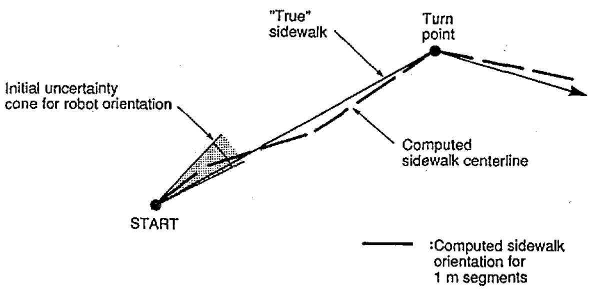


Fig. 3a: Due to difficulties with orienting the robot exactly at the start point, we assume an initial orientation uncertainty of ±10°.

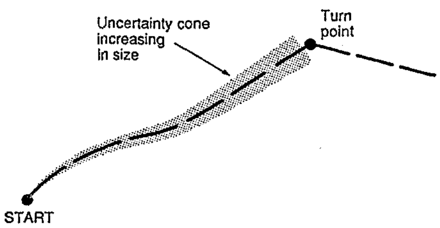


Fig. 3b: Every steering command introduces an orientation uncertainty, primarily due to slippage in the wheels. Since steering commands are computed every 1 meter (unless the distance to the perceived turn-point is less than a meter), we assume an orientation uncertainty of 1° per meter. This causes the orientation uncertainty cone to become increasingly

operator to the digitized gray scale image. These edges are then thinned via Eberlein's algorithm [11] and thresholded. The resulting binary images are thinned again to produce edges that are at most one pixel wide. Any small edges are deleted by the preprocessor. At this point, the image is ready for conversion into symbolic form.

The conversion to symbolic form is accomplished via an algorithm based on the Nevatia-Babu line-finder [12]. In this process, the following steps are performed. First, some pixels are labeled as vertices. The pixels so labeled are edge endpoints and the points at which two or more edges intersect. The edges in the segmented image are then traced from the starting to ending vertices and are represented as broken line segments. After each edge is converted to symbolic form, it contains the following information: edge number, start vertex, end vertex, length and strength (average gradient magnitude). Likewise, each vertex contains the following information: row coordinate, column coordinate, vertex number and degree.

The rule-based labeling system is written in ops5 and is split into three subsystems. The first subsystem is statement driven and does not employ an inexact reasoning scheme. Its purpose is to overcome segmentation deficiencies and reduce the amount of data seen by the section that does use inexact reasoning. There are two main ways that the current segmentation is deficient. First the segmentation procedure leaves small edges caused by noise. Although many of these edges are eliminated during the segmentation process, others still remain because they are connected to longer segments. The first section of the expert system eliminates these "dangling" edges (all segments which are shorter than a specified length and have a degree one vertex). The segmentation

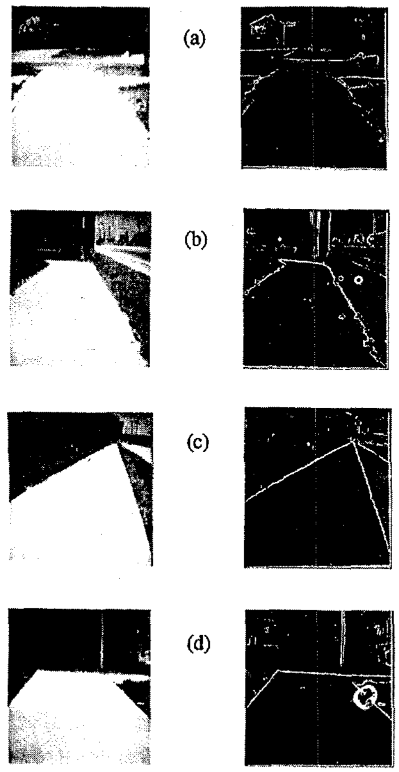


Fig. 4: Shown are the kinds of images that are seen by the downward pointing camera in BAWR mode of navigation. The images on the right, used for drawing inferences about the location and the direction of the sidewalk, were obtained from the images on the left by a sequence of operations described in the text.

used also produces artifacts that break lines into smaller line segments. The system tries to compensate for this fact by rejoining these broken line segments. Finally, the first section combines, into a single line, segments that are joined at a degree-two vertex. The overall result of these subprocesses is a cleaner image containing a substantially reduced number of line segments. Experimental results demonstrate that the amount of pruning is between 66% and 75%.

The second subsystem performs segment labeling and confidence estimation. The system uses the robot's expected position to estimate where it should see the sidewalk's edges in its field of view. (for example, if the robot is expecting the sidewalk to have a right hand turn, it should see the left and right edges for the sidewalk before the turn and the top and bottom edges for the sidewalk after the turn.) The expert system was designed to use the position of expected (model) edges to label those found in the segmented image. The edge labels assigned consist of the following information: the name of the coresponding model edge and a certainty factor between 1 and -1 describing the confidence attached to that labeling. All segments are initially labeled as the model edge to which they correspond most closely.

After the initial labels are assigned, they can not be changed. However, every segment's confidence value is updated based on the consistency between its label and the labels of all other segments and the expected geometry. The belief that a segment's label is correct based on new evidence

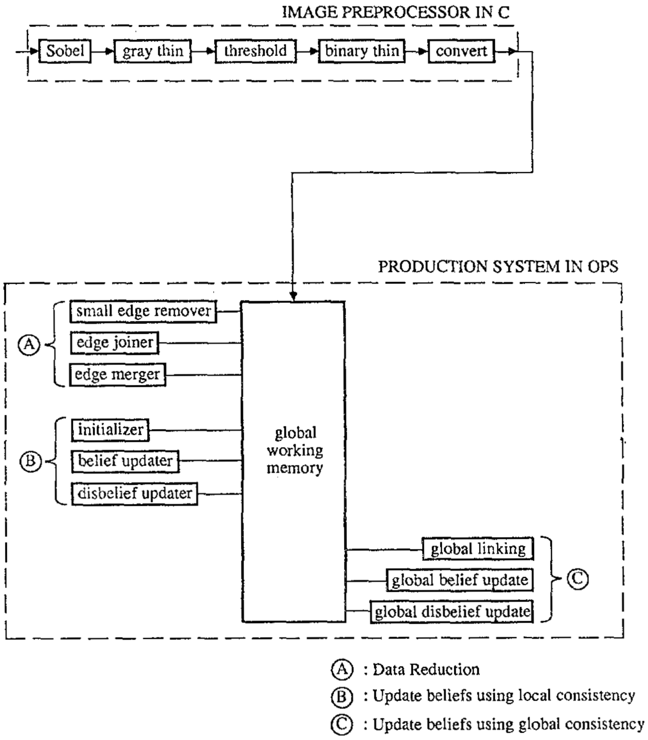


Fig. 5: The architecture of the PSEIKI image understanding system.

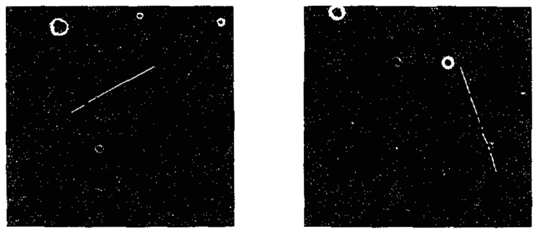


Fig. 6: The left and right edges of the sidewalk in Fig. 4c found by PSEIKI.

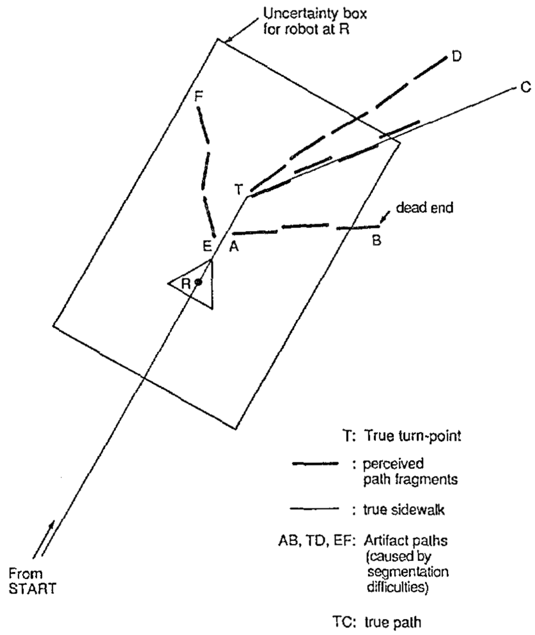


Fig. 7: The figure depicts the robot approaching a turn-point, its current position being such that the turn-point is within the uncertainty box. The robot judgement at R suffers from the fact that the direction along the artifact path fragments AB and TD are within the orientation uncertainty about the true direction TC.

is computed separately from the confidence that it is incorrect (its disbelief). New and old evidence is combined using a variant of the Dempster-Shafer theory of evidence [13]. The following criteria provide evidence that a segment was labeled (in)correctly given another segment's label and confidence value: If two segments are thought to lie on the same model edge, the confidence that the first's label is correct is increased if the segments are highly colinear. If the two segments are believed to correspond to different model edges, a segment's confidence is updated based on how closely the angle between the two segments matches the corresponding model angle. The new belief is defined as the cosine of the difference between the expected and measured angles multiplied by the confidence value of the updating edge. Likewise, the new disbelief is defined as the sine of the difference of the angles scaled by the confidence of the updating edge's label.

The third subsystem used in PSEIKI is not yet complete; it will duplicate the processes used in the first two subsystems but will work on edge groups instead of singletons. Work also must be done to incorporate camera calibration information into the reasoning system; this will remove distortions due to perspective imaging. Figure 6 shows the left and right sidewalk edges found by PSEIKI for the image in figure 4c.

3.4 Interaction of Sensor Errors

The most complex interaction between different kinds of sensor errors occurs in the vicinity of sidewalk turn-points. The interaction begins as soon as a turn-point is within the position uncertainty box of the approaching robot.

In Fig. 7, we have depicted the robot approaching a turn point; its current location being such that the uncertainty box includes the turn-point. By checking its current position with the attributed graph representation of the sidewalks, the robot knows that an upcoming right turn is within the range of positional uncertainty at its present location.

The robot judgement at R also suffers from the fact that the directions along the artifacts path fragments AB and TD are within the orientation uncertainty about the true direction, which *we* know is along TC.

In BAWR mode at position R, the robot with its attendant uncertainties has absolutely no means whatsoever of evaluating the relative quality of the alternative paths, as represented by AB, TC, and TD. Since the robot is expecting to make a right turn, it might as well select AB. In the computer simulation of BAWR presented below, we will discuss how the robot might extricate itself from a wrong path choice.

4. COMPUTER SIMULATION RESULTS FOR BAWR MODE OF NAVIGATION

In this section our simulation of the implementation of the very first level of robot intelligence discussed in section 2, will be presented. The BAWR level of robot intelligence does rely on a global map to provide course navigation/route planning. Fine level navigation is accomplished with the information gleaned from onboard cameras and sonar range sensors. The two or movre onboard cameras transmit their signals back to the lab where the images are subsequently digitized. One of the cameras is pointed slightly under the horizon to provide a long range view of what is ahead and the other camera is pointed toward the ground ahead of the vehicle to provide information about the next 4 meters of travel.

In our simulation of this level, the global map is represented by an attributed graph as shown in Fig. 8. The coordinates of each node, where a node represents a known junction of paths or a turn in a path, are known in advance and are used to provide the graph edges with their distance attribute. Hence when traveling from node 'a' to node 'b' the robot is simply commanded to travel to node b's coordinates relative to node a.

In our simulation, path finding is carried out in Prolog which returns its result in the form of a list of characters. When it is desired to move the robot from node b to node k, a suitable path is represented by the list (baijk). The list contains each node that the robot must visit before arriving at node k.

To simulate vision data, a small box, the vision region, is drawn over the current position of the robot and the section of the bitmap within that box is analyzed. The size of the box varies linearly with the distance from the last node encountered in the planned path. This box size dependence allows for the accumulation of the errors mentioned in section 3.1 and 3.2. When it is determined that the current node of the path list has been reached, all position errors can be set to zero and travel can begin toward the next node of the path list.

Analysis of the vision region must determine whether or not the path ahead is clear and if multiple paths ahead of the robot exist. If multiple paths lie ahead of the robot, the proper path must be discerned. At the BAWR level of intelligence, the robot must travel with the aide of vision guidance along each of the candidate paths to determine if the path does indeed lead the robot in the direction of the destination node. Paths are labeled as candidate paths if the orientation of the robot that is necessary to begin travel down the path is within the orientation error cone. There are 2 principal metrics used in determining if a path is the correct path: 1) the distance between the destination node coordinates and the robot coordinates and 2) the difference between the orientations of the path currently being traveled and the desired path. The orientation of the desired path is known since the coordinates of the destination node are known. The orientation of the path currently under investigation is determined by traveling a fixed distance down the path with the use of vision guidance and then making use of the starting and ending coordinates of the robot as obtained from its dead reckoning position controller. If a path is subsequently declared incorrect, the robot must backtrack to the junction where it made the decision to travel along that path and then repeat the searching process with other candidate paths.

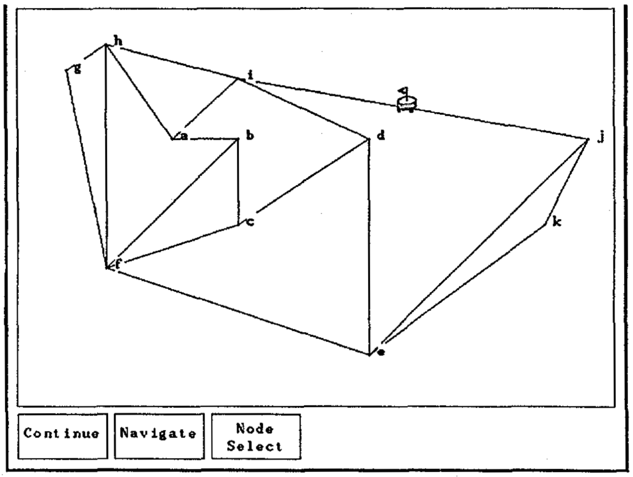


Fig. 8: A sidewalk map is represented in the computer by an attributed graph. So as not to clutter up the figure, only the links and the nodes of a sample graph are shown here.

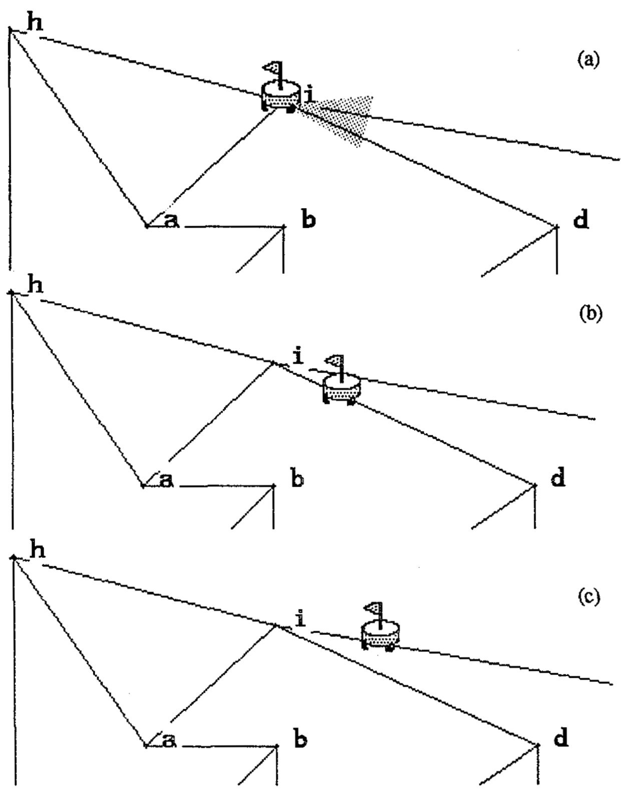


Fig. 9: When the robot makes a wrong path choice at a turn-point (perhaps by getting off the sidewalk and following artifact path fragments generated by segmentation errors), it must backtrack and attempt to get back on the correct path. This figure shows (a) the initial position at a node, (b) the investigation of a candidiate path and (c) investigation of the other candidate path made possible after backtracking off of path i-d.

The sequence in Fig. 9 illustrates the backtracking necessary to deliver the robot to node j given the path list (baij). Note that the paths from node i to node d and the path from i to j fall within the orientation error cone.

Errors in the vision processing have also been simulated by the addition of path artifacts as can be seen in Fig. 10. When path artifacts are the cause of multiple paths, the same blind man backtracking strategy must be employed to determine the true path. Hence path artifacts are treated as true paths until it is determined otherwise. While attempting to travel down a path artifact the vision system will soon fail to discern the path ahead or the sonar system will detect an obstacle in the path. At this point the path must be abandoned and the robot must travel back to the junction at which it decided to travel the path. Upon arrival at the junction the next candidate path is explored and either abandoned or labelled correct or incorrect as described above.

A sensory error occurs when the path to be traveled can not be identified in the vision data. In this case breaks occur in the edges of the path being followed. To correct for this type of error, the images obtained from both of the cameras mentioned above could be correlated to determine whether or not a path truly exists. In simulation when a break occurs in the edge being followed, a large search ring lying within the vision region is used to detect the path on the other side of the break. If the path is detected on the other side of the break, the robot will move ahead as if the path did indeed exist.

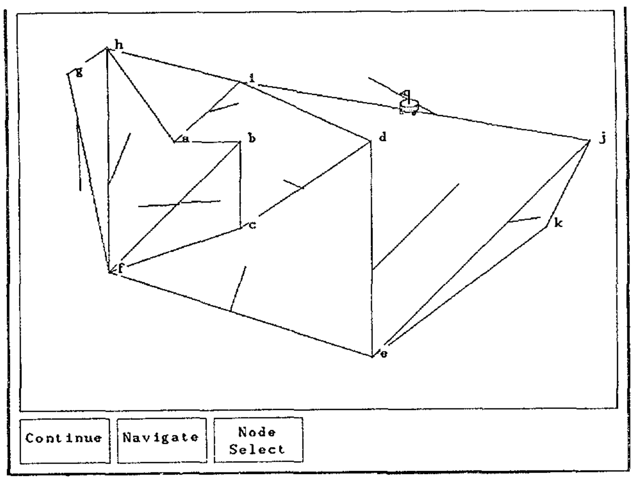


Fig. 10: In our computer simulation of BAWR mode, we also include artifacts along path segments that are between turn-points. These artifacts can be breaks in the sidewalk centerline and the appearance of false path fragments as shown in the figure.

5. CONCLUSION

The main point of this paper is that the level of intelligence of a mobile robot should be determined by the twin considerations of the methods used for world representation, and the methods used for the sensory confirmation of that representation. We proposed three modes of intelligent behavior by a mobile robot; these modes are of increasing sophistication with regard to both the world representation and the methods for integrating sensory information with this representation.

This paper also briefly presented PSEIKI, a Production System Environment for Integrating Knowledge with Images. This system is used for automatic interpretation of the vision data by first assigning initial belief values to edge segments in an image and then updating these belief values by enforcing consistencies at first the local and then the global levels. The initial belief assignments are computed by comparing edge attributes, such as the location and orientation, with important nearby edges in the map representation of a scene. The system uses the Dempster-Shafer formalism for combining belief values.

ACKNOWLEDGEMENT

We are grateful to Matt Carroll, Research Engineer, and Jeffery Lewis, Research Programmer, in the Robot Vision Lab for all their contributions that have helped raise PETER's intelligence from that of an idiot to the level of an imbecile.

REFERENCES

[1] Alberto Elfes, "Sonar-based real-world mapping and navigation," Submitted to IEEE Transactions on Robotics & Automation, June 1986.

[2] Rodney Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics Automat.*, Vol. RA-2, No. 1, pp. 14-23, March 1986.

[3] S. Y. Harmon, "Comments on automated route planning in unknown natural terrain," *IEEE Conf. on Robotics* , Atlanta, GA, pp. 571-573, March 1983.

[4] Toshihiro Tsumura, "Survey of automated guided vehicle in Japanese factory," *Proc. IEEE Int. Conf. Robotics Automat.*, Vol. 3, 1986, pp. 1329-1334.

[5] Tashiro Takeda, Atsuhiko Kato, Toru Suzuki, and Mitsuo Hosoi, "Automated vehicle guidance using spotmark," *Proc. IEEE Int. Conf. Robotics Automat.*, Vol. 2, 1986, pp. 1343-1348.

[6] Alexandre Parodi, John Nitao, Louis McTamaney, "An intelligent system for an autonomous vehicle," *Proc. IEEE Int. Conf. Robotics Automat.*, Vol. 3, 1986, pp. 1657-1663.

[7] Hans P. Moravec, "The Stanford cart and the CMU rover," *Proceedings of the IEEE*, Vol. 71, No. 7, July 1985, pp. 872-884.

[8] James L. Crowley, "Navigation for an intelligent mobile robot," *IEEE J. Robotics Automat.*, Vol. RA-1, No. 1, March 1985, pp. 31-41.

[9] S. V. Nageswara Rao, S. S. Iyengar, C. C. Jorgensen and C. R. Weisbin, "Concurrent Algorithms for Autonomous Robot Navigation in an Unexplored Terrain," *Proc. of IEEE International Conf. Robotics and Automation*, 1986.

[10] K. M. Andress and A. C. Kak, "PSEIKI: A Production System Environment for Integrating Knowledge with images," Technical Report, School of Electrical Engineering, Purdue University.

[11] Robert B. Eberlein, "An iterative gradient edge detection Algorithm," *Computer Graphics and Image Processing*, Vol. 5, 1976, pp. 245-253.

[12] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Computer Graphics and Image Processing*, Vol. 13, 1980, pp. 257-269.

[13] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.