Towards Autonomous Mobile Robot Navigation

C. Fennema, A. Hanson,
E. Riseman

COINS TR 89-104

October 1989

Towards Autonomous Mobile Robot Navigation[1]

Claude Fennema, Allen Hanson, Edward Riseman
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01002

ABSTRACT

The UMass Mobile Robot project is investigating the problem of intelligent navigation of an autonomous robot vehicle. Model-based processing of the visual sensory data is the primary *mechanism* used for obstacle avoidance, movement through the environment, and measuring progress towards a given goal. This paper describes our current approach to goal-oriented navigation through a partially modeled, unchanging environment which contains no unmodelled obstacles.

The navigation system integrates perception, planning, and execution of actions. Of particular importance is that the planning processes are able to reason about landmarks that should be perceived at various stages of plan execution. Correspondence between image features and expected landmark locations are used at several abstraction levels to ensure proper plan execution. Experiments in this and three companion papers demonstrate the performance of the various components within the navigation system.

I. INTRODUCTION

The UMass Mobile Robot project is investigating the problem of enabling a mobile automaton to navigate intelligently through indoor and outdoor environments. At the foundation of our work is the premise that higher-level vision beyond the first stages of sensory processing will greatly benefit from, and in many cases require, the use of knowledge and models of objects in the environment. Thus, model-based processing of the visual sensory data is the primary mechanism used for obstacle avoidance, movement through the environment, and measuring progress towards achieving a given goal.

Our mobile robot, called Harvey, is a Denning platform ultimately intended to navigate through offices, hallways, and university grounds as it carries out commands such as "Fetch the book" or "Bring this to Allen". Since this is a rather formidable task, we have developed a research plan that will be carried out in stages of increasing generality and functionality. In the early phases of this research, we wish to balance generality with setting sufficient constraints on the initial research goals to be achievable. Our initial experiments focus on robust goal-oriented navigation through a partially-modeled, unchanging environment that does not contain any unmodelled obstacles.

If robust autonomous navigation can be achieved in this restricted domain, then a variety of challenging problems can be considered as the constraints are eased on the assumed knowledge about the environment. These problems include: navigation in a partially known environment with obstacles, navigation in the presence of independently moving objects, and exploration of an unknown environment to learn a model in order to support future model-directed navigation. This paper, however, describes the current UMass approach to the initial problem domain of robust navigation in a partially-modelled

[1]This work was supported in part by the Defense Advanced Research Projects Agency under contract numbers F30602-87-C-0140, DACA76-85-C-0008, and DACA76-86-C-0015, and by the National Science Foundation under grant number DCR-8500332.

environment, and our experiments in testing an implementation of such a system.

I.1 Related Mobile Robot Research

We begin with a brief survey of previous mobile robot research; other relevant research will be addressed in the sections discussing particular system modules. The Carnegie-Mellon NAVLAB (Kanade, Thorpe et al. 1986; Shafer, Stentz et al. 1986) and the Martin-Marietta ALV (Lowrie, Thomas et al. 1985) are systems that can move down a path or road or navigate off-road terrain, but the processing has been restricted to simple goals, such as controlling the vehicle relative to the sides of the road, or avoidance of major obstacles such as trees. Recent demonstrations of these systems have been quite interesting, but a laser range sensor providing depth information played a significant role in the obstacle avoidance capabilities.

Brooks (Brooks 1986) has an unusual demonstration of low-level behaviors and motor activity to allow a relatively inexpensive robot to wander in an unknown environment carrying out some purposeful activity, but this work has not yet focused on the achievement higher-level goal-oriented navigation tasks, and does not make use of models of the environment.

Dickmanns and Graefe (Dickmans and Grafe 1988a; Dickmans and Grafe 1988b) have developed techniques for using image features in a real time feedback control loop to control the motion of a car on the autobahn. In the system we develop in this paper their techniques could serve as part of the function we term "action level servoing". The approach described here, like Dickmans and Graefe, accomplishes servoing by tracking image features, but here the tracking features are constructed from landmarks which have been selected from a knowledge base.

Due to the complexity of visual perception, autonomous navigation projects, such as those cited, have utilized only limited visual processing, either in terms of the features extracted from the environment, or the modeled set of objects to be recognized in the environment, or both. This is not meant to be a serious criticism, but rather serves as an observation for the reader who does not recognize the extreme complexity of the problems of vision and autonomous navigation in natural outdoor domains.

Recently, Faugeras (Toscani and Faugeras 1987) used more sophisticated vision algorithms involving stereo to derive depth in an office scene. Depth information was extracted from a stereo pair, the robot was moved some distance, and a second stereo pair was used to derive depth and the associated motion. Again, this effort does not represent a full robot navigation system, and made no use of high-level models.

I.2 Overview of System Modules

The processing modules that provide the basic functional capabilities for our mobile robot system are briefly outlined below. There are many possible control strategies and system organizations that can be imposed on top of these modules to support effective mobile robot navigation. In Section III, we briefly outline one such control strategy.

Modelling the 3D Environment (Connolly 1989; Connolly and Weiss 1989) - Geometer is a solid modelling package that was jointly developed at the University of Massachusetts and General Electric Corp. The CAD system provides tools for representing knowledge of shape in an annotatable hierarchy.

Planning (Fennema, Hanson et al. 1989; Fennema, Riseman et al. 1988)- Tasks (or goals) are translated by a command interpreter and decomposed by a hierarchical problem solver into a sequence of milestones and proposed actions. Plans are developed depth-first, with less detail away from the current task; task failure triggers dynamic replanning.

Monitor Plan Execution (Fennema, Hanson et al. 1989; Fennema, Riseman et al. 1988) - Plans are executed in a repetition of two operations: recognize milestone and execute primitive action. Each milestone is constructed from a perceivable 3D landmark derived from the model. Finding the projection of the landmark in the image signifies a successful completion of the associated action.

2D Line Model Matcher (Beveridge, Weiss et al. 1989a; Beveridge, Weiss et al. 1989b) - This module finds a best match and fit of a given 2D line model to a subset of data line segments that may have been fragmented, skewed, omitted, etc. during low-level processing. A search through the plausible symbolic correspondences between model and data lines is performed, and the optimum 2D translational and rotational fit for each is computed as a closed-form solution.

3D Pose Refinement (Kumar 1989) - Given correspondences between a set of points and lines in a 3D model and a 2D image, the 3D camera location and orientation is computed as an optimization procedure. In addition, uncertainty in the output parameters as a function of the variance of the noise in the input parameters is provided.

In addition to these modules, several basic vision modules have been developed. These modules include a fast line finder (Kahn, Kitchen et al. 1987) derived from a straight line algorithm developed by Burns (Burns, Hanson et al. 1986), a histogram based region segmentation algorithm (Beveridge, Griffith et al. 1989), an algorithm for determining subpixel line placement given an image line, and a local template correlation mechanism (Fennema, Hanson et al. 1989).

II. GEOMETER AND MODELS OF THE ENVIRONMENT

II.1 Geometer

Models of the vehicle's environment are built using Geometer, a three-dimensional solid modeling package developed jointly by UMass and the GE Research and Development Center (Connolly 1989; Connolly and Weiss 1989). Geometer is implemented in LISP and is oriented towards image understanding (although it has many other potential applications). It currently runs on several types of workstations, including the Symbolics LISP machines, TI Explorers, Vax workstations, and Sun workstations. Refer to (Connolly and Weiss 1989) in this proceedings for additional information about Geometer.

Objects in Geometer are represented in an annotatable hierarchy:

World ➡ Object ➡ Faces ➡ Edges ➡ Vertices.

In Geometer, the language of simplicial complexes in algebraic topology (Eilenberg and Steenrod 1952; Greenberg and Harper 1981) has been adapted for describing surfaces. It provides generality and an explicit representation of edges, vertices, and faces. Each of these serve as a type of geometric primitive, and can be parameterized as a smooth function from a point, unit interval, and triangle to R^3 respectively. Surfaces are constructed as the union of these primitives, and are denoted by a sum of simplices. This representation produces a triangulation of the surface, where the triangles are not necessarily planar.

II.2 Constructing Environmental Models

The system begins with an accurate, but incomplete, model of the world implemented in Geometer, augmented by the locale structure described in the next section. We have constructed a 3D model of portions of the interior of the UMass Graduate Research Center, as well as a portion of the campus surrounding the building. The outdoor model (shown in Figures 1 and 2) includes buildings (windows, doors, pillars, etc.), sidewalks, lamp posts, telephone poles, and most of the significant objects in the area. This model has been annotated with properties of objects and surfaces which are useful to the planning and vision routines used by Harvey.

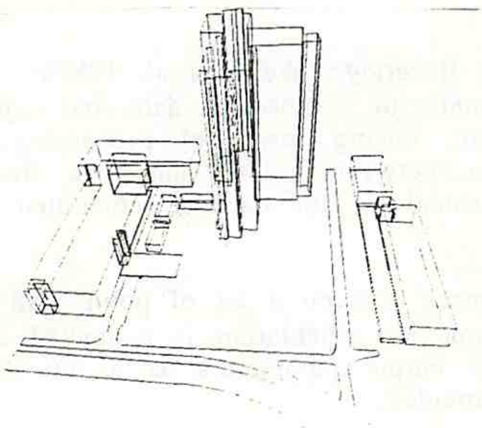

Figure 1. Geometer model of the area around the Graduate Research Center used in the experiments.

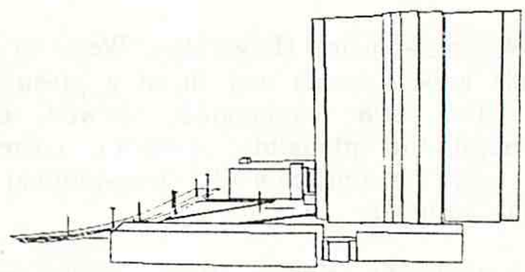

Figure 2. A more detailed Geometer model of the same areas shown in Figure 1 with hidden lines removed. Note that additional landmarks, such as telephone poles, have been added.

The construction of an accurate 3D model of an environment is a fairly difficult job. The first attempt involved digitizing data from engineering blueprints using a bit pad (digitizing tablet). This method is quite error prone given the spatial resolution of the bit pad, since the blueprints were drawn to a scale of 40 feet to an inch We found errors of up to 10 feet in the 3D model constructed in this manner. In the second attempt, theodolites were used to survey the landmarks. This method, while accurate, is very time consuming. As a check, some of the theodolite data was verified by direct measurement. On the average, the measured distances matched with the surveyed distances within 0.2 feet.

II.3 Locales

The model of space in this system plays a rather central role in most of the robot's activities. During planning, for example, the model is used to construct routes. Consequently, the concept of doorways, portals, exits, and entrances must be represented. During plan *monitoring*, the model is used in a top-down fashion to control visual perception by specifying what is to be "seen" and where to "look for" it. In this situation, only the space within the perceptual field of view of the robot is relevant. If the robot gets *lost*, the *world* model is used as a means for localizing it within the environment. Space should be represented and organized in a way which simplifies these tasks.

Conceptually, our view of the organization of space is inspired by the topological notion of a neighborhood. Hierarchically organized neighborhoods serve to successively localize a point to a finer resolution. We use this concept as a means for localizing the agent (robot) by associating with each neighborhood a means for determining whether or not the agent is inside it. This neighborhood-test pair is called a 'locale'. Locales impose an organization on 3D space and partition it into convenient subspaces that are used for planning and robot localization.

A locale is represented by a data structure that captures its neighborhood-like properties via a 3-D shape description of the locale and a contained-by hierarchy as shown in Figure 3. Each locale also contains additonal information, such as its shape descriptors as shown in Figure 4. From this locale data structure, it is possible to construct a test to determine whether or not the agent is in a particular locale and to pick landmarks to act as milestones.

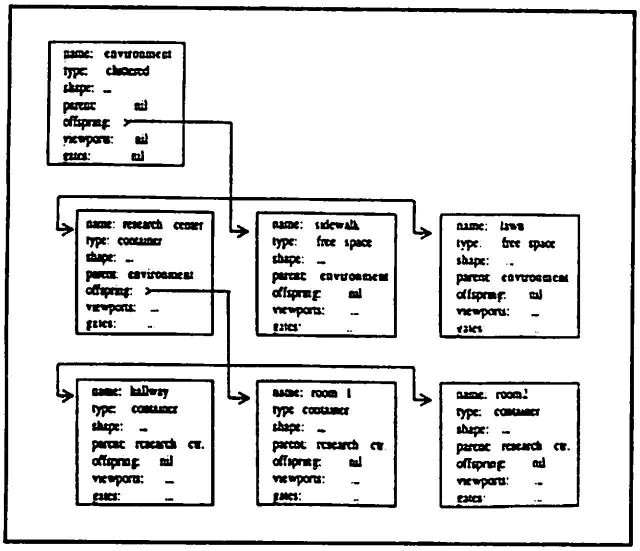

Figure 3. Locales are subspaces of the environment which are organized into a hierarchy by set inclusion. This simplified example shows three levels of locales representing the Graduate Research Center environment.

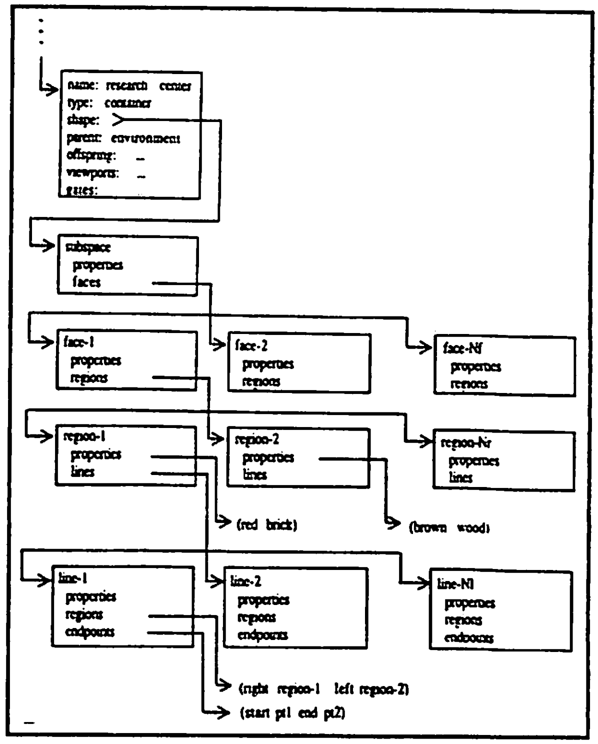

Figure 4. The actual shape descriptions of each locales is a hierarchy of geometric entities defined in Geometer. Shown are the entity properties used during perceptual reasoning to construct landmarks.

III. A BRIEF LOOK AT PLANNING AND CONTROL

Each task given to Harvey is translated by a command interpreter and problem solver which ultimately produces a set of navigational goals. The execution of these goals is accomplished by a tight interweaving of planning, perception, and action, orchestrated by a dynamic planning and execution scheme (Fennema, Hanson et al. 1989; Fennema, Riseman et al. 1988) called "plan-and-monitor". This subsystem works with plans, each represented as a sequence (M0 A1 M1 ... AG MG) of milestones (Mk) and proposed actions (Ak). Milestones are used to verify the successful completion of a particular phase of the plan. They are composed of 3D landmarks (perceivable physical events) and their expected location with respect to the robot at the completion of the appropriate phase of the plan.

As a plan is executed milestones must be verified (usually visually) before the next action of the plan can be executed. For example, if the sequence of milestones up to M7 have been perceptually verified to be in the proper position in the image (i.e. within the acceptable error bounds), this means that actions A1, ..., A7 have been successfully completed, and it is appropriate to take action A8. If M7 cannot be verified, then the plan must be modified. In this way milestones allow the progress of the plan to be monitored, and trigger replanning before the next action is taken when perception and milestone do not agree(Fennema, Riseman et al. 1989). Complex actions and tasks also trigger replanning in order to refine them into a plan subsequence of milestones and primitive actions which can be directly executed by the hardware.

The plan-and-monitor executive directs planning, perception, and execution in such a way as to dynamically modify and refine the plan to fit the actual results of each action and the details of the perceived environment. The principal activities involved in this process are: planning, milestone recognition, determination of location, and execution of primitive actions. This interweaving of perception, planning and action makes specific what task is expected of perception, and provides a means for focusing the knowledge available for that purpose. The results is a distribution of perception and perceptual reasoning into all aspects of navigation. Route planning uses perceptual reasoning to select appropriate perceptual milestones; plan progress is measured using perception; perception is used to relocate the robot when a milestone is not recognized; and during the execution of primitive actions, low-level perceptual feedback is used to keep the robot on the expected trajectory. The different levels of control all use model-directed vision and compare what is sensed to what is expected, issuing corrective commands to minimize any difference.

Plan execution depends upon the recognition of milestones. The difficulty of using vision to perform this task in a reliable and general manner has encouraged us to attack this problem in two ways. Both methods use model-directed processing by comparing restricted perceptual processing to what is expected if the robot's motor actions are correct. The next section describes a type of low-level perceptual servoing used during execution of primitive actions. Section 5 describes a more complex method for matching models to landmarks and refining the position of the robot based on these matches.

IV. EXECUTING PRIMITIVE ACTIONS: PERCEPTUAL SERVOING

Navigation goals are ultimately translated into primitive actions which can be directly executed by the robot vehicle; in the case of the Denning platform, these are (MOVE distance) and (TURN angle). Even at this primitive action level, however, execution errors are probable. As the robot rolls along an environmental surface a slippery spot, a bump on the surface, or even a bulge in its tire may throw it off course, causing inaccurate execution.

It is possible to reduce the error incurred when executing a primitive action by *servoing* on prominent visual features in the environment. Using information obtained from the measured discrepancy between where the features should be and where they actually are, it is possible to determine the corrective action required to bring the positions into agreement. This *action level* or *perceptual servoing* has the effect of locking the robot onto a trajectory which improves the accuracy of the primitive actions over that which would be obtained without servoing.

In order to determine the usefulness of servoing, a simple version was implemented that used correlation to measure the deviation of actual motion from intended motion. Several experiments, both with and without correlation servoing, were run. In the experiments Harvey was to roll along a straight line 40 feet long, marked on the floor of a Graduate Research Center hallway For the experiments in which servoing was used, an artificial target was placed on a door at the end of the hallway, since the Geometer model of the interior of the building was not complete. The target was a circle approximately eight inches in diameter with two opposing black quadrants and two opposing white quadrants. The robot's goal was to move down the corridor directly towards the target. To determine course deviation, the vehicle was stopped every two feet and its deviation from the marked line was measured.

The experiment was run a number of times; the results in Table 1 represent the best one in the sense that the unservoed results represent the smallest deviations encountered during the trials. The z-axis referred to in the table is the line the vehicle is following with z=0 defined as the starting location. The x-axis is a line perpendicular to the z-axis and pointing to the right. Total distance traveled is z-unservoed and deviation is x-unservoed. Even after a rather painstaking set up procedure the vehicle wandered over two inches from the line during a 20 foot motion. Other runs resulted in as much as a foot deviation in unservoed mode. Most of the trials in unservoed mode were stopped at around 20 feet because the vehicle was significantly off course and the total deviation was increasing. In contrast, in servoing mode the vehicle stayed within .3 inch of the line for 38 feet. It is worth noting that in both experiments the actual distance covered was considerably less than the intended distance. It is consistently short by a constant factor (to 3 decimal places), due to inaccurate calibration of the hardware.

Table 1. Results from one experiment
(All measurements are in inches)

intended-z	unservoed-z	servoed-z	intended-x	unservoed-x	servoed-x
24.	22.6	22.8	0.0	0.0	+0.13
48.	45.5	45.7	0.0	-0.3	+0.13
72.	68.3	68.5	0.0	-0.4	+0.13
96.	90.9	91.3	0.0	-0.6	+0.13
120.	114.2	114.3	0.0	-0.7	+0.06
144.	136.3	136.9	0.0	-1.1	0.00
168.	158.2	159.5	0.0	-1.3	-0.13
192.	181.8	182.2	0.0	-1.8	-0.13
216.	204.6	205.0	0.0	-2.0	-0.38
240.	228.3	227.7	0.0	-2.1	-0.25
---.	---.-	---.-	-.-	-.-	-.--
480.	---.-	456.0	0.0	-.-	0.0

The results of these experiments are encouraging and support the idea of action level perceptual servoing over reasonably short navigation legs; additional results for (MOVE distance) as well as servoing results for (TURN angle) are presented in (Fennema, Hanson et al. 1989). Once the Geometer model of the building interior is complete, similar experiments will be performed using actual geometric features rather than the artificial target. When weather permits, the vehicle will be moved outdoors and the Geometer model described in Section II will be used to determine the effect of terrain cover and topography on servoing accuracy.

V. RECOGNIZING AND USING 3D LANDMARKS

Recognition of 3D landmarks involves matching an object model to data extracted from an image, and this task has two parts: a)determining the correct correspondence between object features and image features and, b)determining the position of the object with respect to the camera. We refer to the former task as 2D model matching (Section V.1) and to the latter as 3D pose refinement (Section V.2) These sub-tasks are interdependent, since an object's position relative to the camera in 3D space cannot be determined without determining a correspondence to image features, while the correct correspondence depends on the object's 2D appearance and hence its relative position and orientation in space.

V.1 2D Model Matching

In contrast to the approach developed by Lowe for the SCERPO system (Lowe 1985; Lowe 1987) we have chosen to separate the 2D processing of model-to-image matching from the 3D optimization process for computing the camera pose once the correspondences between model and image are completed. Thus, we restrict ourselves in this section to the problem of matching a 2D model to a set of fragmented, skewed, and missing line segments, a rather challenging perceptual organization problem. The model line to image line correspondences determined from this 2D matching method are used as the input to the 3D pose computation discussed in the next section.

We believe that there are strong incentives to solve as much of the identification problem as possible via processing in the 2D image space. The combinatorics of establishing correspondences between object and image features dominates the identification problem, and geometric computations integral to this process are simpler in 2D than in 3D. In particular, Beveridge et al (Beveridge, Weiss et al. 1989a; Beveridge, Weiss et al. 1989b) show that the determination of the optimal position of an object's 2D projection with respect to corresponding line features has an analytic solution in the two dimensions of image space. This closed form solution for line correspondence is a new result and we believe it to be a significant contribution. It is highly doubtful that the related 3D problem has an analytic solution for determining model positions that minimize point-to-line and point-to-plane distances.

Given that matches will seldom if ever be perfect, the emphasis must be on determining the 'best' of the imperfect matches. Hence matching is naturally posed in terms of optimization over the possible matches. By establishing an objective measure of match quality, the problem becomes one of determining the correspondence between model elements and data line segments for which the measure is optimal. The correspondence problem is combinatorial, and generally involves mapping one model line to many data lines. A second optimization problem is implicit in the correspondence problem. In order to measure the quality of a given model-data line correspondence, the best 2D position of the model with respect to the data must be determined, and the extent to which they do not spatially coincide must be measured. This we call the *fitting* problem. Hence, a match involves both model-data correspondence and an associated best-fit position.

The following is a sketch of the basic steps used to obtain a good model match:

- Determine the search space of correspondences. Lacking constraints on model position, all data lines segments possibly correspond to every model line segment. If constraints are available, only pairs of model and data lines satisfying these constraints need be considered.

- Determine promising model positions if the search space is large. Use these positions to determine constrained search subspaces made up of only correspondences consistent with the estimated position. A promising model position may be found either through a generalized hough transform or by identifying prominent features. The generalized hough technique involves an analysis of the space of possible two-dimensional spatial transforms to bring the model and data into alignment. Identifying a prominent feature may involve finding a distinctive part of a model, such as a corner and then using that to position the model as a whole.

- For each of the constrained search spaces obtained above, use iterative refinement to determine a best match. Upon each iteration perturb the correspondence, adding or deleting one or several data lines, and then determine the new best-fit model position and related match error. If the match error is reduced adopt the improved match. Stop when the match can no longer be improved. The best of the resulting matches is taken as the final match.

Results are presented for 2D model matching in Beveridge (Beveridge, Weiss et al. 1989b) using both synthetic data and images obtained from the robot vehicle. Sample results from this paper for one frame of a six frame image sequence is shown in Figures 5 and 6. The output from the 2D model matching system provide the input for the 3D pose refinement computation presented in the next section.

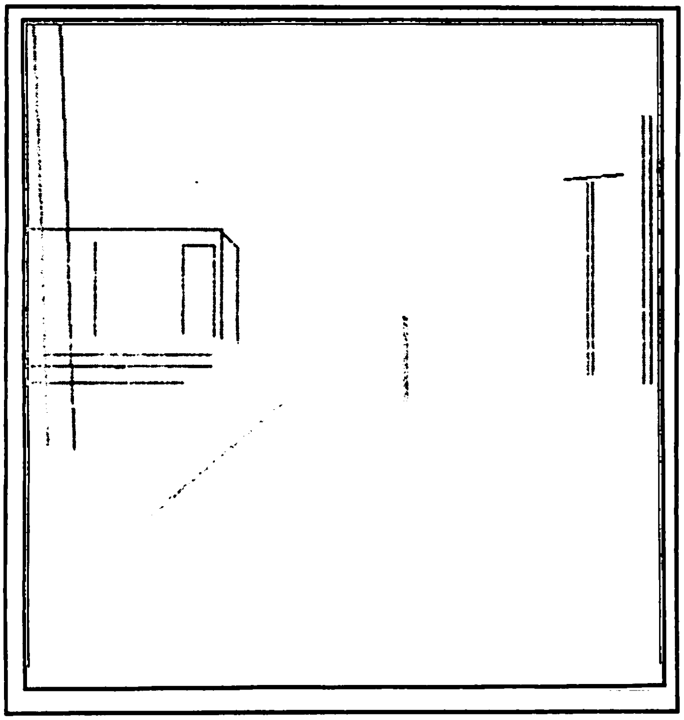

Figure 5 2D Modeling Matching Results. Projections of the six 3D navigation landmark models onto the 2D image plane using the current position of the robot.

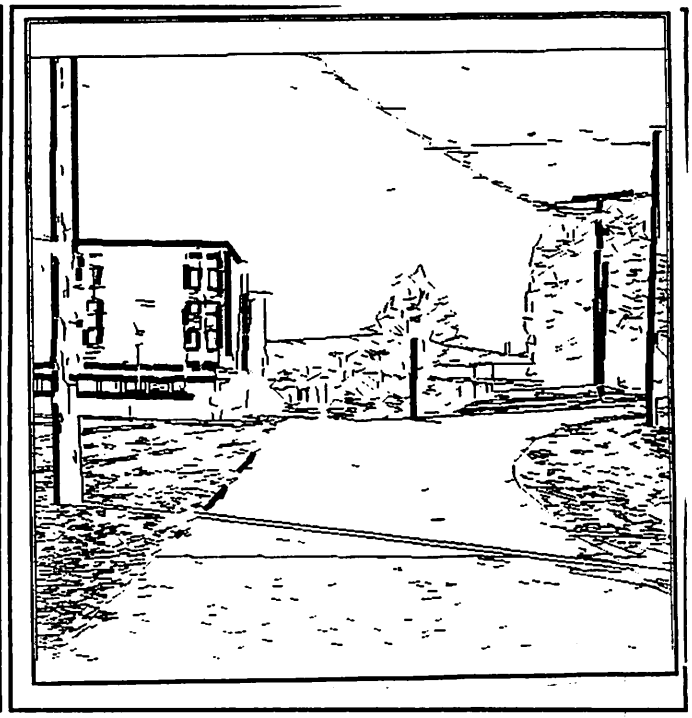

Figure 6. 2D Model Matching Results. Matches of model line segments with image line segments; the dark lines represent the matches. These matches are used by the 3D pose refinement module described in Section V.2

V.2 3D Pose Refinement

Kumar (Kumar 1989) develops a solution to and mathematical analysis of the problem of estimating camera location and orientation from a set of recognized landmarks appearing in the image. Given correspondences between the 3D landmark model lines and 2D image lines, the goal is to find the camera (or robot) rotation and translation which map the world coordinate system to the camera coordinate system under perspective projection. Because of the difficulties encountered in trying to establish accurate endpoint positions for lines (Kumar 1989; Lowe 1985; Williams and Hanson 1988), we assume that correspondences established between model and data are line correspondences and not endpoint correspondences. In addition, intrinsic camera parameters, such as focal length, field of view, center of the image, size of image, etc. are assumed to be known (Horn 1986; Kumar 1989; Lenz and Tsai 1988).

This problem, under various names and guises, has been addressed by several researchers, e.g. see (Ganapathy 1984; Horn 1987; Linnainmaa, D. et al. 1988; Wolf 1974); most of the techniques assume line endpoint data, are iterative in nature, and require an initial estimate. Liu, Huang, and Faugeras (Liu, Huang et al. 1988) present a solution to the "camera location determination" problem which works for both point and line data. Kumar's approach is based on their constraints, derived from the observation that the 3D lines in the camera coordinate system must lie on the projection plane formed by the corresponding image line and the optical center. Using this fact, Liu et. al. separated the constraints for rotation from those of translation, leading to a solution in which rotation is solved for first and then translation is obtained, using the rotation results.

The technique developed by Kumar to solve for the rotation and translation parameters differs from that of Liu et al in two significant ways. First, rotation and translation are solved for simultaneously, which makes more effective use o the constraints and is more robust in the presence of noise. Second, the nonlinear technique used to solve for rotation and translation is adapted from Horn (Horn 1987) Kumar's version of this optimization technique provides much better convergence properties than does Liu et al's solution method based on Euler angles.

Kumar also develops uncertainty measures for the rotation and translation parameters. Noise in the data is assumed to be only in the image. The 3D model data is assumed accurate. The data for each image line can be specified by two parameters θ_i and ρ_i (a polar coordinate representation of lines). For the analysis, the noise for both θ_i and ρ_i is assumed to be Gaussian distributed with zero mean and known variances. Furthermore, the noise is assumed to be uncorrelated for different lines. Closed form expressions are developed for the variance of the error in the output parameters (rotation and translation) as a function of the input data and output translation and rotation values. Kumar shows that the error in the output parameters is linearly related to the noise in the input data. The reader can refer to Kumar's paper in these proceedings for more details.

Figure 7 shows the results for one frame (the same frame shown in Figures 5 and 6) of the six frame sequence used in one of the experiments. The figure shows the 3D model lines after projection back into the image plane using the vehicle "pose" computed by the 3D pose refinement algorithm which solves simultaneously for the rotation and translation parameters. For this particular frame, the errors (in feet) for the position of the robot (x,y,z) are (.1, 06, .03); additional results for the other frames of this sequence are given in Kumar's paper in these proceedings (Kumar 1989).

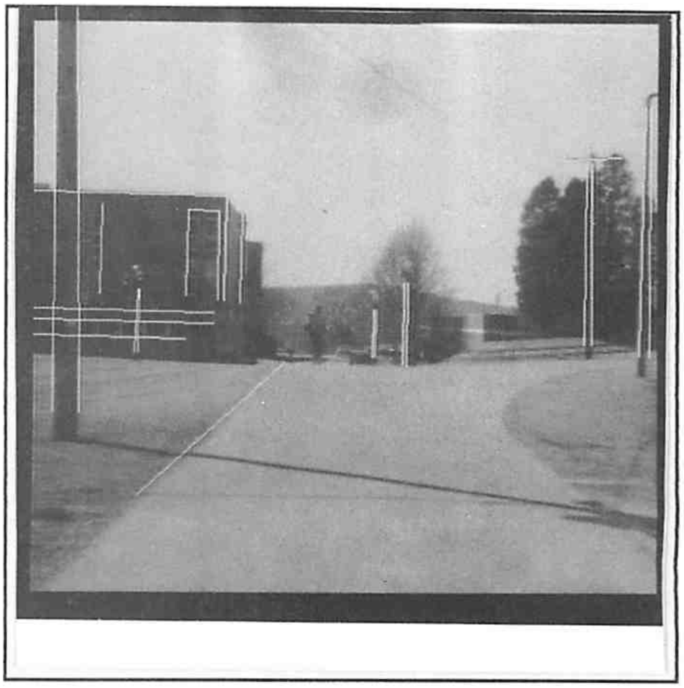

Figure 7. Results from 3D Pose Computation. The white lines are the 3D landmark segments reprojected onto the image plane after 3D pose refinement using the model line-data line matches shown in Figure 6.

VI. CONCLUSIONS

The work presented here represents the current status of a long term research effort leading to the development of perceptually-based navigation systems for autonomous robots. The focus of the research is on environmental modeling, planning, plan monitoring, and vision. These four components are tightly coupled in a system which provide the flexibility and extensibility required for an experimental testbed for robot navigation.

Because the vagaries of the physical world affect plan execution in unknown ways, plans, no matter how carefully constructed, cannot simply be blindly executed. Each step of the plan must be carefully monitored and compared to expectations. The system accomplishes this by defining milestones associated with each planned action. The milestones act as preconditions for subsequent plan steps; the next step cannot be executed unless the milestone is satisfied. This assures a correspondence between the environmental model and the assumed position of the robot relative to the model and the actual position of the robot in the physical world. Failure to satisfy a milestone causes replanning to take place. Interweaving perception, planning, and action in this way makes specific what task is expected of perception and provides a means for focusing available knowledge on local goals.

Experimental results from the system thus far are encouraging, although a number of issues remain to be explored. Harvey's world is completely known, which is perhaps an unrealistic assumption for an autonomous robot. The perceptual servoing mechanisms assume that 3D landmarks can be accurately extracted from the geometric model of the environment. It remains to be seen how the requirement of complete knowledge can be relaxed yet still maintain the idea of perceptual servoing. Incorporating the type of reasoning demonstrated by the schema system (Draper, Brolio et al. 1989) might allow Harvey to respond to instructions like "....continue down North Pleasant street past the Graduate Research Center, then turn left and..."

A unique feature of the model matching component is the separation of the process of positional updating into two steps: 2D matching followed by 3D pose refinement. The robustness of this technique must be determined and its computational efficacy over many experiments in multiple domains must be explored.

Finally, navigation is an extremely computationally demanding task, yet real-time performance is crucial for a mobile automaton whose survival may depend upon reaching critical decisions in a short period of time. An ongoing aspect of the work reported here is the exploration of means by which the navigation task may be distributed over suitably configured parallel architectures. Two complementary lines of research are currently underway, utilizing a Sequent Symmetry multiprocessor system and the University of Massachusetts Image Understanding Architecture.(Weems, Levitan et al. 1989).

VII. References

Beveridge, J. R., J. Griffith, R. Kohler, A. Hanson and E. Riseman. (1989). "Segmenting Images Using Localized Histograms and Region Merging," IJCV, Vol. 2(3), pp. 311-347.

Beveridge, J. R., R. Weiss and E. Riseman. (1989a). "Matching and Fitting Sets of 2D Line Segments to Broken and Skewed Data," Dept. of Computer and Information Science, University of Massachusetts (Amherst), TR In preparation.

Beveridge, J. R., R. Weiss and E. Riseman. (1989b). "Optimizing Two-Dimensional Model Matching," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA.

Brooks, R. (1986). "A Robust Layered Control System for a Mobile Robot," IEEE J. of Robots and Automation, Vol. RA-2(1), pp. 14-23.

Burns, J. B., A. Hanson and E. Riseman. (1986). "Extracting Straight Lines," IEEE PAMI, Vol. 8(4), pp. 425-455.

Connolly, C. (1989). "Geometer: Solid Modelling and Algebraic Manipulation," Dept. of Computer and Informtion Science, University of Massachusetts (Amherst), TR In Preparation.

Connolly, C. and R. Weiss. (1989). "Geometer: A System for Modelling and Algebraic Manipulation," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA.

Dickmans, E. and V. Grafe. (1988a). "Applications of Dynamic Monocular Machine Vision," Machine Vision and Applications, Vol. 1, pp. 241-.

Dickmans, E. and V. Grafe. (1988b). "Dynamic Monocular Machine Vision," Machine Vision and Applications, Vol. 1, pp. 223-240.

Draper, B., J. Brolio, R. Collins, A. Hanson and E. Riseman. (1989). "The Schema System," IJCV, Vol. 2(3), pp. 209-250.

Eilenberg, S. and N. Steenrod. (1952). Foundations of Algebraic Topology, Princeton, NJ: Princeton University Press.

Fennema, C., A. Hanson, E. Riseman, R. Beveridge, R. Kumar, C. Connolly and R. Weiss. (1989). "Model Directed Mobile Robot Navigation," Dept. of Computer and Information Science, University of Massachusetts (Amherst), TR in preparation.

Fennema, C., E. Riseman and A. Hanson. (1988). "Planning With Perceptual Milestones to Control Uncertainty in Robot Navigation," Proc. of SPIE -- International Society for Photographic and Industrial Engineering, Cambridge, MA.

Fennema, C., E. Riseman and A. Hanson. (1989). "Planning with Perceptual Milestones to Control Uncertainty in Robot Navigation," Proc. of AAAI Spring Symposium Series (Working Notes: Robot Navigation), Stanford University, pp. 19-23.

Ganapathy, S. (1984). "Decomposition of transformation matrices for robot vision," Proc. of 1st IEEE Conference on Robotics, pp. 130-139.

Greenberg, M. and J. Harper. (1981). Algebraic Topology: A First Course, Reading, MA: Benjamin.

Horn, B. K. P. (1986). Robot Vision, Cambridge, MA: MIT Press.

Horn, B. K. P. (1987). "Closed-form solution of absolute orientation using unit quarternions," J. Opt. Soc. A., Vol. 4, pp. 629-642.

Kahn, P., L. Kitchen and E. Riseman. (1987). "Real-Time Feature Extraction: A Fast Line Finder for Vision-Guided Robot Navigation," Dept. of Computer and Information Science, University of Massachusetts (Amherst), TR 87-57.

Kanade, T., C. Thorpe and W. Whittaker. (1986). "Autonomous Land Vehicle Project at CMU," Proc. of ACM Computer Conference.

Kumar, R. (1989). "Determination of Camera Location and Orientation," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA.

Lenz, R. and R. Tsai. (1988). "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology," IEEE-PAMI, Vol. 10(5), pp. 713-719.

Linnainmaa, S., H. D. and L. Davis. (1988). "Pose determination of a three-dimensional object using triangle pairs," IEEE-PAMI, Vol. 10(5), pp. 634-647.

Liu, Y., T. Huang and O. Faugeras. (1988). "Determination of Camera Location From 2D and 3D Line and Point Correspondences," Proc. of CVPR, Ann Arbor, MI, pp. 82-88.

Lowe, D. (1985). Perceptual Organization and Visual Recognition, Boston, MA: Kluwer Academic Publishers,.

Lowe, D. (1987). "The Viewpoint Consistency Constraint," IJCV, Vol. 1(1), pp. 58-72.

Lowrie, J., M. Thomas, K. Gremban and M. Turk. (1985). "The Autonomous Land Vehicle (ALV) Preliminary Road-Following Demonstration," Proc. of Intelligent Robots and Computer Vision (SPIE 579), D.P. Casasent, Ed., pp. 336-350.

Shafer, S., A. Stentz and C. Thorpe. (1986). "An Architecture for Sensor Fusion in a Mobile Robot," Proc. of IEEE International Conference on Robotics and Automation.

Toscani, G. and O. Faugeras. (1987). "Structure from Motion Using the Reconstruction and Reprojection Technique," Proc. of IEEE Workshop on Computer Vision, pp. 345-348.

Weems, C., S. Levitan, A. Hanson, E. Riseman, J. Nash and D. Shu. (1989). "The Image Understanding Architecture," IJCV, Vol. 2(3), pp. 251-282.

Williams, L. R. and A. R. Hanson. (1988). "Translating Optical Flow into Token Matches and Depth from Looming," Proc. of 2nd International Conference on Computer Vision, Tarpon Springs, FL, pp. 441-448.

Wolf, P. (1974). Elements of Photogrammetry, New York: McGraw Hill.