Design of ARK, a sensor-based mobile robot for industrial environments

S. B. Nickerson[1], M. Jenkin[2], E. Milios[2], B. Down[3], P. Jasiobedzki[3], A. Jepson[3], D. Terzopoulos[3], J. Tsotsos[3], D. Wilkes[3], N. Bains[4], K. Tran[4]

[1]Ontario Hydro Research Division, 800 Kipling Ave., Toronto, Ontario, Canada
[2]York University, 4700 Keele St., North York, Ontario, Canada
[3]University of Toronto, Toronto, Ontario, Canada
[4]Atomic Energy Canada Ltd, Mississauga, Ontario, Canada
[2] Fax: 416-736-5872, e-mail: {jenkin,eem}@cs.yorku.ca

Abstract

We describe the design of ARK (Autonomous Robot for a Known Environment), a visually guided mobile robot for navigating in an industrial environment whose major structures have been previously mapped. ARK uses visual landmarks for navigation, the position of which are indicated on the robot's map. The position estimation problem is solved through the use of an active vision sensor, equipped with a video camera and a laser ranging device, which is used for detecting and tracking landmarks. Since it is likely that some areas are not adequately covered by landmarks, ARK plans paths which minimize a weighted sum of path length and path uncertainty. The global path planner assumes that the robot will use a Kalman filter to integrate landmark information with odometry data to correct path deviations as the robot moves, and then uses this information to choose a path which reduces the expected path deviation. As a result, a longer path may be selected, if more landmarks are visible from it, thereby helping the position estimation process.

1 Introduction

ARK (Autonomous Robot for a Known environment) is a visually guided mobile robotics project undertaken as a collaboration between Ontario Hydro, Atomic Energy Canada Limited (AECL), The National Research Council of Canada (NRC), and two universities; York University and the University of Toronto. The goal of this project is to build an autonomous vehicle capable of performing inspection and monitoring tasks in a typical industrial environment. The project will eventually construct two different robots; a tethered robot, called ARK-1, which will be used as a proof of concept vehicle, and a second generation vehicle, called ARK-2, which will utilize the best of the results of ARK-1 and which will operate in real time with the majority of its computation performed onboard. Both robots will use visual data obtained through active vision processes as the primary source of sensing for the robot, but both robots will also utilize non-visual sensors such as infrared, sonar and laser range finders.

The design of the ARK-1 robot (which will be referred to as ARK) has been driven by the final application domain of ARK-2. The ARK-2 robot will operate in a complex industrial environment, similar to the types of environments found in industrial bays, power plants, and other large complex manufacturing centers. These environments are radically different from the lab environments typically used in robotic research. In industrial environments the lighting varies dramatically from location to location and even by time of day. There are a number of permanent structures in the environment (external walls, pillars, etc.), but there are also a number of large semi-permanent structures (manufacturing cites, mockups, etc.), and even temporary structures. In addition to the complexity of the global environment, the local environment is considerably more complex than that found in most robotic research labs. The floor may contain cable trays, drainage ditches, and the like. Objects may protrude at arbitrary heights and from arbitrary orientations. Walls, in the classic sense, may only be found at the environments external boundaries.

It is not practical in the ARK-2 environment to perform modifications of the environment such as adding bar codes to the walls, magnetic strips beneath the floor, or active radio beacons, and as an industrial environment lacks the wall structure of an office or lab space, navigation by necessity has to rely primarily on landmark detection, identification and tracking [8,5,16,17,14]. Thus the design of a sensor capable of localizing landmark features and algorithms for acquiring those features is crucial to the performance of the robot. In addition, the global path planning activities of the robot rely on this active vision sensor. This paper presents the basic design of the ARK vision sensor and describes a path planning procedure which takes the knowledge of the active vision sensor and the landmark localization procedure into account when performing global path planning.

2 The ARK Active Vision Sensor

The ARK-1 robot is built around a 3 wheeled Cybermotion K2A platform, which has been augmented with additional sensors which are used primarily for local path planning (see [19] for a general description of the ARK project). One of the most novel features of the ARK robot is its use of active vision to localize visual cues in the robot's environment.

Active Vision[2] is a research paradigm inspired by the structure of biological vision systems[3], and has been shown to have benefits for a number of visual tasks[22]. To implement this paradigm in practice, a particular experimental apparatus is required to provide control over the acquisition of active image data. One approach considered by a number of researchers has been the construction of robotic "heads" which provide mechanisms for modifying the geometric or optical properties of the sensors under computer control. Several research groups have built stereo robotic heads which rely purely on visual data to recover structure in the environment (for example, see [4,13,1,21,6,18]). Rather than relying purely on visual data, the ARK active vision head combines a spot laser sensor (manufactured by Optech) with a computer controlled zoom and focus color video camera. This combined sensor of video and range data is mounted within a pan and tilt unit providing the robot with the ability to point the combined sensor in different directions (see Figure 1).

The Optech laser range finder obtains depth measurements by temporally integrating individual infrared laser range measurements. The beam spread is quite small with a 5 mrad beam divergence[20]. The sensor operates in one of two modes, a slow accurate mode (0.55 sec/scan) with an accuracy of $\pm$5cm and a faster slightly less accurate mode (0.12 sec/scan). The sensor has an operating range of 0.2 to more than 100m. Note that unlike stereo robotic heads, the range error associated with the laser scanner is independent of distance. The laser is also eye safe making it suitable for use in inhabited environments.

3 ARK Control Structure

The ARK robot has two different control structures. A low level reactive control system whose major goal is to prevent the robot from damaging itself or its environment, and a higher level control structure which allows

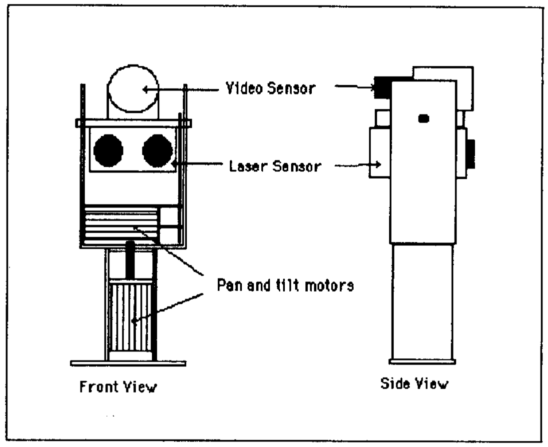

Figure 1: The ARK robot head. At the top is the zoom and focus controlled lens, and at the bottom is the Optech laser range finder. The pan-tilt unit is operated by two DC servocontrolled motors from Micromo Electronics, equipped with gearboxes and optical encoders.

user defined missions to be carried out. These missions include tasks such as "move the robot to its recharge point". The higher level control structure performs operations such as path planning and then utilizes the reactive control system to carry these tasks out.

The low level system is based around the onboard computing available on the K2A platform. The K2A platform integrates a number of specially designed sonar sensors and a bumper with a drive subsystem which can be used to prevent the robot from driving into obstacles which appear in its path. In addition to the functionality provided by the Cybermotion base, additional operations related to the operation of the active vision head are also defined.

One fundamental operation provided by the high level system is that of path planning and visual landmark selection. By taking the need for visual landmark sensing into account when planning paths the high level control structure attempts to plan paths which will allow the robot to move with certainty throughout its environment.

4 Global path planning

ARK takes an approach to path planning which incorporates a number of novel ideas. It integrates (i) the use of sensory information to use landmarks in the environment to correct odometry errors when the robot moves, (ii) knowledge of this integration process to choose paths which reduce the maximum uncertainty in the path chosen, and (iii) a tradeoff between the shortest paths that avoid obstacles and paths that are exposed to landmarks, and hence help the position estimation process.

Constructing an environmental potential field

Although the ARK robot utilizes a hybrid representation for its map, it is sufficient to consider a simpler occupancy grid representation here.

The basic approach in an occupancy grid based path planner is to superimpose a grid over a blueprint of the robot's workspace. Each cell in the grid is assigned a certainty that it contains a solid object. The process of converting a blueprint into certainty measurements has been addressed by a number of authors. Very sophisticated path planning problems can be solved using this type of representation. For example, Hwang and Ahuja [10], have developed a simulation-only system which can solve problems involved in moving complex pieces of furniture through very small openings. As in [10] structure in the environment generates a potential field. Each grid element is assigned a potential field value equal to the maximum potential field that reaches this cell. The use of the maximum, rather than an additive field is to avoid the generation of local maxima in the field. In order to treat the robot as a point for planning purposes, the robot's footprint is treated as a circle and then all obstacles are thickened by this radius. Similar to the approach of Hwang and Ahuja, ARK uses a potential field function given by

$$p(\vec{x}) = \max(\delta + \mathrm{dist}(\vec{x}, o_i))^{-1} \qquad (1)$$

where o_i is the closest point on the i'th object to $\vec{x}$, and the index i runs over all of the known structure in the environment. The potential field is basically $1/d$ where d is the distance to the closest obstacle. The term δ is simply a small positive number used to prevent overflow situations. As the map used by ARK is built up of two dimensional geometric primitives, the task of determining the closest obstacle is quite straightforward.

Estimating the robot's position

For any robot navigation process to operate, the robot must be able to relate knowledge about the environment (the map) to information obtained with the robot's sensors. In particular the robot must be able to know where and with what orientation, the robot is on the map. To simplify the process, it will be assumed that the robot can be modelled as a point robot, capable of straight line motions and rotations about the robot's current position. Of course, in order for any of this to work the robot has to know where it is with respect to the map. For simplicity, we assume that the robot starts out in a known state and this is established through a registration procedure. Using only the laser range finder, two registration procedures have been implemented. The first [7] is based on point to line-segment matching, and seeks a pose correction (translation and rotation) that minimizes the total squared distance between each data point and the line segment in the map that is closest to it. Data points that are outliers are detected as having too large a distance from every line segment of the map, and they are discarded from the least squares formulation of the problem. The second [9] seeks a pose correction that minimizes the total squared distance between point features in the map and in the sensed range data. A variety of point features can be used, such as convex or concave corners, or occlusion boundaries (points), that can be reliably detected in the range data and also unambiguously defined in the map. The current implementation uses occlusion points, which can be identified precisely by using a bisection method similar to that for solving non-linear equations and the pan-tilt ability of the ARK robot head. In a manner similar to the treatment of outliers in the point-to-line matching method, we associate with eath data point the model point that is closest to it. Data points that are too far away from every model point are considered outliers and they are not included in the least squares minimization formulation. The advantage of the point-to-point least squares problem is that it has a closed form solution, unlike the point-to-line version, which must be solved iteratively. The disadvantage is that its accuracy depends on the accuracy with which the point features are known in both the data and the map, and is probably more sensitive to such errors than the point-to-line version.

For integrating odometry with sensor information, perhaps the most successful technique so far has been the use of Kalman filtering. A number of different robots have successfully utilized this technique including Kleeman [11] and Leonard and Durrant-White [15,16]. The Kalman filter provides a prediction of where the landmarks should be in the next image, thus facilitating position estimation. The Hough transform has also been used in addition to a Kalman filter to limit the search for matching edge elements [12]. The ARK robot also utilizes a Kalman filter to integrate sensory information with odometry information to provide a least squares estimate of the robot's position. Unlike the more common sonar based case, ARK utilizes visually localized pre-mapped landmarks as the sensory input to localize the robot. ARK is expected to navigate primarily via its active vision sensor. This sensor (mounted on some point on the robot) obtains the distance and direction (from the sensor) to a known landmark on the map. For navigational purposes, we can assume that this orientation and distance are two dimensional. In practice the landmarks are located at different altitudes and the sensor will actually obtain three dimensional direction and distance measurements. This information is used for target identification, but we consider only the more simple case of two dimensional data here. In a measurement cycle, the robot consults the map to identify landmarks which may be visible from the robot's current position. The active vision sensor is then used to locate any landmarks that may be visible. Zero, one or more landmarks may potentially be visible at any point, and the active vision sensor will return the measured direction and distance to each of the identified landmarks.

5 Path planning with uncertainty

As Kalman filtering is to be used to estimate the robot's position on its path it is possible to estimate the covariance matrix associated with the robot's state before physically moving the robot. Suppose that before physically moving a robot down a path, the robot's motions are simulated using the robot's map of its environment and position of the landmarks. In this simulation it will be assumed that (i) the robot always goes to where it is expected to go, and (ii) that the sensors always report the correct measurement, then even though the simulated robot will always go to where it is supposed to, and there are no measurement errors, it will still be possible, via the Kalman filter equations, to compute the state covariance matrix $\vec{P}(k|k)$ at each step in the robot's path. Thus for each step in each path it is possible to estimate (given that the robot and sensors perform flawlessly), the uncertainty of a particular step and path. Define

$$cost(p_i) = \beta \sum_{i}^{n} P_{field}(p_i) + (1 - \beta) \max_{i} Trace(\vec{P}(i|i)) \quad (2)$$

to be the cost of a path, where $P_{field}(p_i)$ is the occupancy field measure at step i in the path, and $Trace(\vec{P}(i|i))$ is the Trace of the Kalman covariance matrix at step i. This is equivalent to the sum of the Eigenvalues of $P(i|i)$ which is a measure of the size of the uncertainty region at the i'th step. Minimizing *cost* is a trade off (controlled by β) between paths which minimize the certainty grid cost (weighted path length) and paths which minimize the maximum positional uncertainty over their length. Paths which minimize *cost* are both short and sure.

The potential field used to represent the free space of the robot defines a metric array representation of the position of the robot. This space is expanded to include a discrete representation of the orientation of the robot (currently 8 different orientations to match the rectangular decomposition of the position space). Thus the possible positions of the robot (for path planning) are discretized both in position and orientation. Each position has a potential field value associated with it. For each step in the path planning process, the robot may move no farther than an adjacent cell in position, and may change orientation arbitrarily. Given a starting position, and a goal position, an A^* algorithm (or Djikstra's algorithm) can be used to find a path from the start to the goal which minimizes the path cost function given in (2).

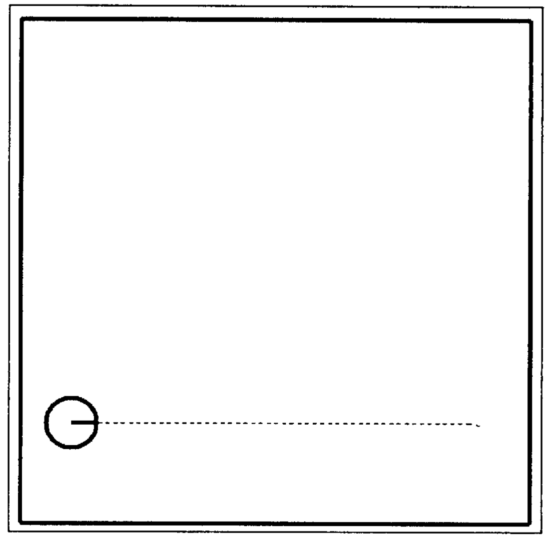

Figure 2: Shortest distance path planning. The initial pose of the robot is shown as a circle and the planned path is indicated by a dotted line.

6 Testing

Although the uncertainty based path planner operates in very complex environments it is more instructive to consider its application in a very simple environment such as an enclosed rectangular room with no internal obstacles. Figures 2, 3, and 4 show different paths planned by a robot in moving from one point to another in an empty environment. The paths planned by the robot are shown as dotted lines. Figure 2 shows the path planned by a shortest path algorithm. The robot plans the straight path from the start to the goal. Figure 3 shows the path planned by an occupancy grid minimizing algorithm. As the centre of the room is farther away from obstacles (the walls), the best path involves moving towards the centre of the room before moving towards the goal.

Figure 4 shows the same environment but has the addition of a visible landmark at the top of the room (indicated by a small circle). The visibility region of this landmark is shown as a semicircular region centred on the landmark. The same start and goal positions are used as in Figure 2 and Figure 3. The planned path moves away from the goal position towards the visibility region of the landmark in order to reduce the robot's uncertainty and then moves back towards the goal after moving through the landmark visibility region for some distance.

In addition to planning the robot's path, the Kalman uncertainty based path planner records the landmarks that were used to compute the positional uncertainty of the robot at each point in the path. These planned sightings of landmarks are also displayed as dotted lines

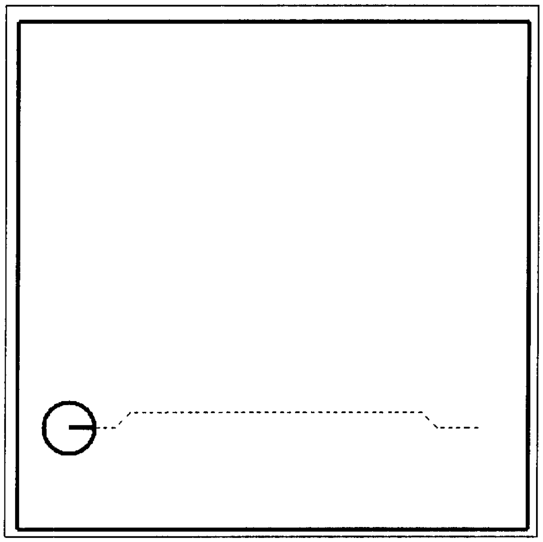

Figure 3: Occupancy cost path planning. The initial pose of the robot is shown as a circle and the planned path is indicated by a dotted line.

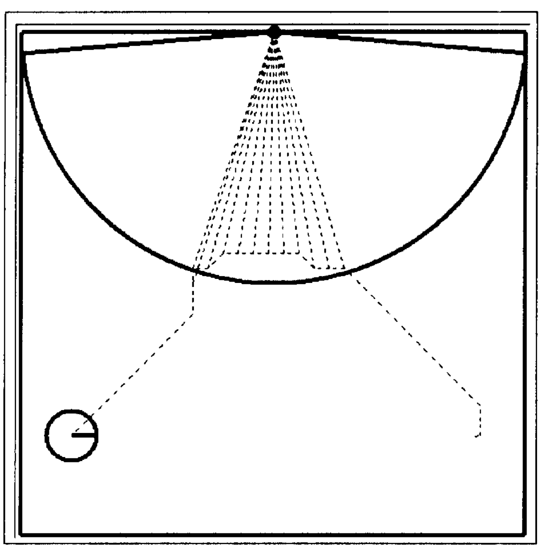

Figure 4: Uncertainty based path planning. The initial pose of the robot is shown as a circle and the planned path is indicated by a dotted line. Expected landmark sightings are also shown as straight lines from the robot's path to the landmark.

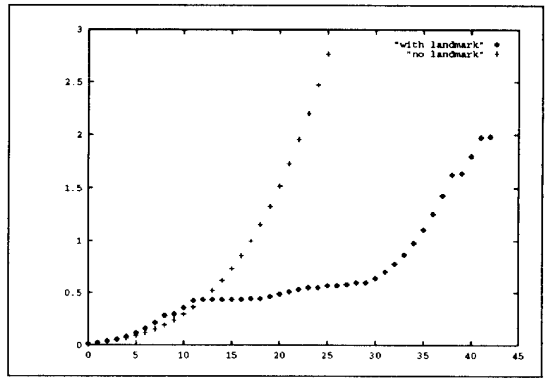

Figure 5: Uncertainty path cost. The maximum uncertainty cost at different points in the path is shown for the planned path with (displayed in figure 4) and without (displayed in figure 3) the presence of the landmark. As is clearly evident the longer path has a much lower maximum uncertainty than the shorter one.

from the robot's planned path to the landmark position. When the path is finally executed this information is used to prime the ARK head to acquire a specific landmark at an approximate position for position estimation.

The uncertainty-based path planner can also be used if no landmarks are present in the environment. In this case the planner computes a path which is similar to the distance minimizing path. For the environment presented here the path computed by the Kalman path planner if the landmark is removed is exactly the same path as the distance minimizing path shown in Figure 2. In addition to computing the path the expected Kalman covariance matrix is also computed. Plotting the costs of the uncertainty based planned path with and without the presence of the landmark is very informative (see Figure 5). The plot shows the maximum Trace of the covariance matrix at each point in the path for the path planned with landmarks and the path planned without. The path with landmarks is longer but has long regions of low uncertainty (corresponding to moving within the visibility region of the landmark). The path planned without the landmark is shorter but has a much higher positional uncertainty at the goal. For some environments it may be worthwhile to move a farther distance to be more sure of getting to the goal.

7 Discussion

The ARK robot navigates through the use of visual landmarks acquired with an active vision sensor. This sensor is a computer controllable laser-color video pair. This combination of sensors allows for active visual search for landmark features through controllable changes in focal length, pan, tilt and focus. The head also has the ability to make spot depth measurements with the laser sensor. Premapped visual landmarks are identified with the sensor and foveated. The direction to the landmark and its distance are used by the robot to correct the positional errors associated with the robot's drive system. Knowledge of this correction process and the relative sparseness of visual landmarks suggests the use of path planner which approaches path planning by identifying paths which are likely to be navigatable with the sensors, and sensor integration mechanisms currently on board. Rather that identifying short or safe paths, paths identified with this planner will be suited to the onboard sensors, rather than being independent of them.

Many enhancements to the ARK global path planner are possible. It would be possible, for example, to discard any path with positional uncertainties that exceed a particular value (hence identifying locations that can be reliable accessed). The path planner currently does not utilize any heuristics to suggest growing the search space towards the location of the known goal. This could be easily added. Finally, even the walls themselves can be thought of as landmarks for navigation, and it would be straightforward to add additional classes of landmarks (and their visibility regions) into the path planner.

Acknowledgements

Funding for this work was provided, in part, by the ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., the Department of Industry, Science and Technology Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro, and Atomic Energy of Canada Limited. Financial support from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

[1] A. Abbott. *Dynamic integration of depth cues for surface reconstruction from stereo images.* PhD thesis, Electrical Engineering, Univ. of Illinois at Urbana-Champaign, 1990.

[2] R. Bajcsy. Active perception vs. passive perception. In *Third IEEE Workshop on Vision*, 1985.

[3] D. Ballard. Eye movements and visual cognition. In *Workshop on Spatial Reasoning and Multisensor Fusion*, pages 188–200. Morgan Kaufmann, 1987.

[4] D. Ballard and A. Ozcandarli. Eye fixation and early vision: kinetic depth. In *International Conference on Computer Vision*, pages 524–531, 1988.

[5] R. Beveridge and E. Riseman. Can Too Much Perspective Spoil the View? A case study in 2D Affine versus 3D Perspective Model Matching. COINS TR91-86, Computer and Information Science Dept., University of Massachusetts at Amherst, 1991.

[6] F. Chenavier and J. Crowley. Position estimation for a mobile robot using vision and odometry. In *IEEE Int. Conf. on Robotics and Automation*, pages 2588–2593, 1992.

[7] I. J. Cox. Blanche - an experiment in guidance and navigation of an autonomous robot vehice. *IEEE Trans. on Robotics and Automation*, 7(3):193–204, 1991.

[8] C. Fennema, A. Hanson, E. Riseman, J. R. Beveridge, and R. Kumar. Model-directed mobile robot navigation. *IEEE Trans. on Sys. Man and Cyber.*, 20(6):1352–1369, 1990.

[9] W. E. L. Grimson. *Object recognition by computer: The role of geometric constraints.* MIT Press, 1990.

[10] Y. K. Hwang and H. Ahuja. A potential field approach to path planning. *IEEE Trans. on Robotics and Auto.*, 8(1):23–32, 1992.

[11] L. Kleeman. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *IEEE Int. Conf. on Robotics and Automation*, pages 2582–2587, 1992.

[12] A. Kosaka and M. Meng and A. Kak. Vision-Guided Mobile Robot Navigation Using Retroactive Updating of Position Uncertainty. In *IEEE Int. Conf. on Robotics and Automation*, Vol. II, pages 1-7, 1993.

[13] E. Krotkov. *Active computer vision by cooperative focus and stereo.* Springer-Verlag, New York, 1989.

[14] B. Kuipers and T. Levitt. Navigation and mapping in large-scale space. *AI Magazine*, pages 25–43, 1988.

[15] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Auto.*, 7(3):376–382, 1991.

[16] J. J. Leonard and H. F. Durrant-Whyte. *Directed sonar sensing for mobile robot navigation.* Kluwer Academic Publishers, 1992.

[17] I. Masaki, editor. *Vision-based vehicle guidance.* Springer-Verlag, New York, 1992.

[18] E. Milios, M. Jenkin, and J. Tsotsos. The design and performance of trish, a binocular robot head with torsional eye movements. *Int. J. of Pattern Recognition and AI*, 7(1), 1993. (to appear).

[19] S. B. Nickerson, D. Lond, M. Jenkin, E. Milios, B. Down, P. Jasiobedzki, A. Jepson, D. Terzopoulos, J. Tsotsos, D. Wilkes, N. Bains, and K. Tran. Ark: Autonomous navigation of a mobile robot in a known environment. In *IAS-3*, pages 288–296, 1993.

[20] Optech Systems Corporation, 701 Petrolia Rd, North York, Canada. *Model G150 Rangefinder User Manual*, 1991.

[21] K. Pahlavan and J-O. Eklundh. A head-eye system - analysis and design. Technical report, Computational Vision and Active Perception Laboratory, Royal Institute of Technology, S-100 44 Stockholm, Sweden, October 1991.

[22] J. K. Tsotsos. On the relative complexity of active versus passive visual search. *Int. J. of Comp. Vis.*, 7(2):127–141, 1992.