An autonomous mobile robot for known industrial environments

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

[1] Ontario Hydro Technologies, Toronto, Canada
[2] Atomic Energy of Canada Ltd., Mississauga, Canada
[3] Dept. of Computer Science, York University, North York, Canada
[4] Dept. of Computer Science, University of Toronto, Canada

This draft printed August 28, 1997

Abstract

Although mobile robot navigation in unstructured environments is an open problem, robotic systems can be developed which operate in relatively unstructured but known environments. The Autonomous Robot for a Known environment (ARK) project has constructed an autonomous robot capable of navigating and carrying out survey/inspection tasks in a complex industrial environment. The ARK environment is an industrial factory floor, and therefore lacks the planar walls and well-defined corridors of a typical office environment, where the majority of laboratory mobile robots operate. The ARK robot relies on naturally occurring objects localized with *Laser Eye*, a novel active vision sensor, as visual landmarks for navigation. This paper describes the overall structure of the ARK project, the technical results that it achieved, and the robotic systems it produced.

1 Introduction

There are many types of industrial operations and environments for which mobile robots can be used to reduce human exposure hazards, or to increase productivity. Examples include inspection for spills, leaks, or other unusual events in large industrial facilities, materials handling in computer integrated manufacturing environments, and the carrying out of inspections, the cleaning up of spills, or the carrying out of repairs in the radioactive areas of nuclear plants - leading to increased safety by reducing the potential radioactive dose to workers. It is this industrial survey and inspection task in that the ARK (Autonomous Robot for a Known Environment) project addresses.

Many industrial environments are highly instrumented in order to diagnose anomalous conditions and to allow for a rapid response to them. Unfortunately, the instrumentation itself is fragile and a considerable amount of time and money must be expended in responding to failures of the instruments and their communication mechanisms. Thus one potential application of mobile robotics in an industrial environment is to act as an independent verification of existing instrumentation. For this type of robotic application to be effective, it must be possible to direct the robot to a specific location described in a global metric coordinate system and to instruct the robot to verify the function of the suspect sensor. In order to accomplish this task it is thus essential that the ARK robot know its location at all times with respect to an *a priori* global map of the environment.

The industrial environment is significantly different from the office environments in which most mobile robots operate. The test environment for the ARK robot is the large engineering laboratory at AECL CANDU in Mississauga, Ontario. This open area covers approximately 50,000 sq. feet of space and accommodates one hundred and fifty employees. Within the Laboratory, there are test rigs of various sizes, mockups of reactor components, a machine shop, a fabrication facility, a metrology lab and assembly area. There are no major barriers between these facilities and therefore at any one time there may be up to fifty people working on the lab floor, three fork lift trucks and floor cleaning machines in operation. Such an environment presents many difficulties for a mobile robot including: the lack of vertical flat walls; large open spaces (the main isle is 400' long) as well as small cramped spaces; high ceilings (50'); large windows near the ceiling resulting in time dependent and weather dependent lighting conditions, a large variation in light intensity, also highlights and glare; many temporary and semi-permanent structures; many (some very large) metallic structures; people and fork lifts moving about; oil and water spills on the floor; floor drains (which are sometimes uncovered); hoses and piping on the floor; chains hanging down from above, protruding structures, and other transient obstacles to the safe motion of the robot[18]. Figure 1 shows the industrial prototype ARK-2 robot in the AECL industrial bay.

Large distances, often encountered in an industrial environment, require sensors that can operate at such ranges. The number of visual features (lines, corners and regions) is

Figure 1: The ARK industrial prototype robot within it's operating environment.

very high and techniques for focusing attention on specific, task dependent, features are required. Most mobile robotic projects assume the existence of a flat ground plane over which the robot is to navigate. In the industrial environment this ground plane is generally flat, but regions of the floor are marked with drainage ditches, pipes and other unexpected low lying obstacles to movement. To operate in an industrial environment, a robot requires sensors that can reliably detect such obstacles, algorithms to move the robot and maintain its position within the environment, and control algorithms that allow the robot to operate safely in spite of the existence of other moving entities within the environment.

The ARK robot must navigate through its environment autonomously and cannot rely on modifications to its environment such as the addition of beacons[22], magnetic strips beneath the floors[13], or the use of visual symbols added to the existing environment. The ARK robot must rely on objects which occur naturally within its environment as landmarks. As many of these existing landmarks are visual in nature, the robot relies on vision as its main sensor for global navigation, using a map of permanent structures in the environment to plan its path.

In addition to a set of technical goals, the ARK project was required to meet a set of industrial goals. In order to meet ongoing performance reviews it was essential that the project develop a prototype system in stages. In addition to allowing the project to develop the robotic system in an incremental fashion, the early deployment of a prototype allowed researchers a realistic hardware environment within which more advanced code could be developed.

(a) ARK-1

(b) ARK-2

(c) ARK-lite

Figure 2: The ARK robots. The ARK-1 robot is based on the Cybermotion Navmaster platform and is shown here with a commercial pan and tilt unit upon which is mounted the active vision sensor Laser-Eye described in the text. The ARK-2 Cybermotion platform has been heavily modified through the addition of on-board processing and additional sensors and power. ARK-lite is based on the Nomad 200 platform modified through the addition of a computer controlled pan and tilt unit.

2 The ARK robots

The ARK project constructed three robotic prototypes (see Figure 2). At the University of Toronto, ARK-1 was used as an initial testbed on which the ideas, sensors and algorithms were tested that were ultimately included in ARK-2, the industrial prototype. For ARK-1, computation was primarily performed off-board using standard workstations, while ARK-2 utilizes special purpose real-time computers and most of the computation is performed on-board. A second research machine, ARK-lite, was installed at York University. All three robots use visual data obtained through active vision processes as the primary source of sensing for the robot. They also use non-visual sensors such as infrared, sonar and laser range-finders. ARK-1 and ARK-2 are based on the Cybermotion Navmaster platform, while ARK-lite is based on the Nomad 200.

The main hardware components of the ARK-1 robot are a Navmaster mobile platform from Cybermotion, and a robotic head with sensors and a remote link to a host computer network. The platform consists of a base with three synchronous drive wheels and a rotating turret. The Navmaster comes equipped with a contact sensitive bumper and six sonars, two of them facing forward, two backward and two sideways. Additional sonar sensors are mounted on the turret or the bumper to enhance the interpretation of the sonar data (see [29]). The ARK-1 robot communicates with a network of host computers via an 8-channel remote serial link. The communication between the robot and the host is on the level of processed signals from sensors and commands sent to the robot. The on-board computers collect the data from various sensors, pre-process it and send it via a radio link to the host computer network. The computers in the network analyze this data, and generate commands for individual units of the robot (platform, head, sonar controllers, range-finder). The on-board computers perform time critical functions such as emergency stop, positioning the head and moving the platform. The host network of computers is based on standard Unix workstations. This arrangement is particularly convenient for software development but it does make it difficult to experiment with real-time responses to external events. The non-real-time nature of the Unix operating system combined with unpredictable delays in the serial modem conspire against real-time control on ARK-1.

In ARK-2, the vast bulk of the computation, such as processing and interpretation of data from various sensors and generation of control commands, is performed on-board. The communication link in ARK-2 is based upon a wireless Ethernet link which has a much higher bandwidth than is available with the serial link on ARK-1. In addition, ARK-2 is equipped with a wireless video link which runs independently of the wireless Ethernet. The wireless link on ARK-2 is used primarily for exchanging messages between the robot and an operator. The on board computer operates under control of a real-time operating system (VXWorks).

ARK-lite provides a small amount of on-board computation, with more complex computation being processed off-board via general purpose workstations. Off board communication is provided via a spread-spectrum Ethernet link, while a video camera mounted on a pan and tilt unit, and bumper, infrared, and sonar sensors are also available on-board the robot.

3 Project Overview

The primary operational task of the ARK robots is to perform sensing/survey operations within an industrial environment with respect to a global metric map. The application task and operating environment define the envelope within which the ARK robots were developed. An analysis of the requirements of the final system identified a number of key constraints;

1. It is not acceptable to modify the robot's operating environment. From an industrial point of view, fixed beacons or markers to assist in the navigation of the mobile robot make the navigation system fragile, since its ability to perform will require regular maintenance of the markers. This consideration eliminates solutions which rely on the addition of markers, beacons, or guide-paths to the environment.

2. At all times the robot must be able to determine its position with respect to a global metric map of its environment. This requirement arises from the system's need to

be able to direct the robot to particular locations defined within a global coordinate system.

Given the constraint that the environment cannot be modified, the ARK robot relies on the use of existing pre-mapped visual landmarks to correct errors in odometry and hence to provide global navigation with respect to a metric map. Subsequent surveys and preliminary testing within the test environment for the robot yielded many potential candidates for visual landmarks. Typical landmarks within the AECL laboratory consist of alpha-numeric location signs, fire extinguisher markers, doorways, overhead lights, and pillars. The only criteria used for selecting landmarks is that they are distinguishable from the background scene by colour or contrast. These criteria allow the use of both grey level and colour image processing algorithms for landmark identification.

3. The robot must operate in a safe manner. It must be able to react in an intelligent manner to unexpected and unmodelled obstacles and events within its environment.

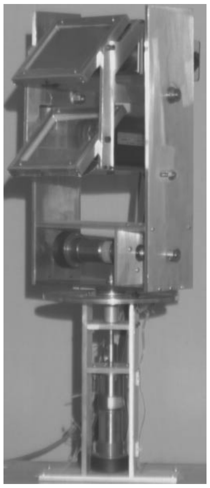
   It is thus essential for the robot to have effective sensor coverage of the environment and to be able to react to external events in an efficient and effective way.

4 Sensing for pose maintenance

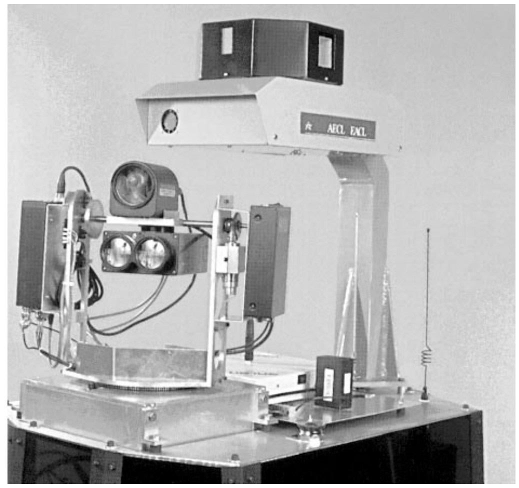
Given the incremental errors associated with odometry, mobile robots require references to external objects in order to accurately maintain their position with respect to a global map. We have experimented with different techniques to use the visual measurements to correct the robot's global position including, as well as more *ad hoc* pose update algorithms. Vision alone is a poor mechanism for constraining the pose of the robot based on sightings of distant landmarks. Although the azimuth and elevation of a landmark can be used to determine distance to a landmark, the computation is not always robust especially for targets near the altitude of the sensor. Thus in order to improve the performance of the pose maintenance process, a special-purpose combined vision and distance sensor was constructed for the ARK robots.

4.1 Combined Vision / Range Sensor

Given the constraints within which the ARK robot must operate and the need to have an accurate estimate of the robot's position at all times, a special-purpose sensor was constructed to acquire the visual landmarks upon which pose estimation would be based. A novel laser/vision sensor *Laser Eye*[4] was designed as the main navigation sensor for the ARK-1 and ARK-2 robots. This sensor provides colour images and a single range measurement to distances up to 100m which are typical for the industrial environment.



(a) University prototype



(b) Industrial Unit

Figure 3: Laser-Eye - The robot head with a combined vision and range sensor. Three different versions of Laser-Eye were eventually constructed. Version 1 was based on a commercial pan and tilt unit and did not organize the laser and camera so that the axes were coincident. Figure 2(a) shows this sensor mounted on ARK-1. (a) above shows the first university prototype with coincident axes. (b) above shows the production unit mounted on ARK-2.

Laser Eye is a combined range / video sensor consisting of a camera and a laser range-finder[8]. The range-finder uses the time-of-flight principle and provides a single depth measurement for each orientation of the sensor. Measuring distances to objects in the scene requires pointing the sensor at each in turn and reading their depth. The range-finder uses an infra-red laser diode to generate a sequence of optical pulses that are reflected from a target. The time required to travel to and from the target is measured to estimate the distance.

Laser Eye has four degrees of freedom: two extrinsic - head pan and tilt, and two intrinsic - camera zoom and focus (see Figure 3). The head can tilt in any direction between 65^o below and 95^o degrees above the horizon and the panning range covers 360^o. The head can rotate with speeds exceeding 180^o/sec. The initial prototype was used on ARK-1 and in early experiments on ARK-2. A commercial version of the head was constructed at AECL and appears on ARK-2 in Figure 3b.

The range-finder within Laser-Eye measures distance to an object in the centre of the camera field of view. In the university version of Laser-Eye the camera optical axis and

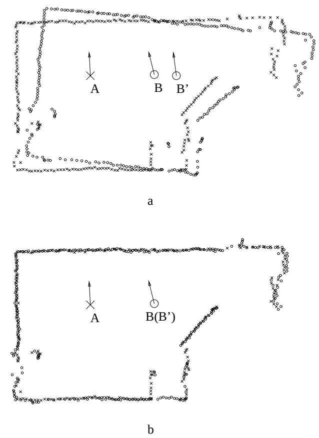


Figure 4: Alignment of laser scans to obtain the initial robot pose. (a) shows the superposition of two laser scans, one from a known position and a second from an unknown position. (b) shows the superposition after application of the correction algorithm.

t of the range-finder were made coincident using a hot mirror (one that reflects infra-red and transmits visible light) placed in front of the camera lens. The mirror transmits the visible light from the observed scene to the camera with minimum attenuation. The hot mirror reflects the transmitted infra-red beam and sends it in the direction of the optical axis of the camera. The returning pulse is reflected by the hot mirror again and projected on a detector in the range-finder[8]. A single range measurement takes 0.12 - 0.5 seconds depending on the selected accuracy. The time required to point the head in a new direction depends on the required rotation. The laser beam divergence is less than 5 mrad. This corresponds to a laser spot of 3 pixels in diameter for an image digitized in a 512x512 grid and for the wide setting of the zoom lens (45^o). For the narrow setting of the zoom lens (4.5^o) the spot is 30 pixels in diameter.

4.2 Pose maintenance with Laser-Eye

Different techniques have been used by each of the ARK robots to exploit the features of Laser-Eye for various pose maintenance tasks.

4.2.1 Initial pose estimation

Perhaps the most primitive pose maintenance task is that of obtaining an initial pose of the robot when it is first powered on. As this process is only performed at the start of a mission or when the normal pose maintenance process has failed, the on-line requirement of the pose maintenance task is avoided and more time-intensive processes can be considered. One technique that was found to be very effective in environments with significant wall structure is the use of the time of flight laser coupled with an *a priori* wall model or scan from a known position.

Figure 4 shows the superposition of two laser scans obtained with Laser Eye in the research labs at the University of Toronto. Given a scan from a known location, a second scan from an unknown location and initial guess of the robot's "unknown" position, a non-linear, robust statistical technique [14] is used to obtain a pose correction to minimize the error between the recovered scan and known wall positions. This process is relatively slow as each data point takes on the order of 0.5 seconds to obtain, but obtains very accurate estimates of the robot's pose. This process is effective as the laser measurement noise process is very well behaved as there are very few surfaces which are specular to the laser in the environment. A novel method for optimally aligning not just two, but a larger number of range scans obtained from different robot positions is presented in [15].

4.2.2 Landmark recognition and tracking

In normal operation, landmark recognition and subsequent measurement of azimuth and elevation towards detected landmarks is the main mechanism for maintaining the ARK robots on their course. ARK-1 and ARK-2 explored different approaches of solving the landmark recognition problem. ARK-2 used a generalized template matching technique of grey-scale images, while ARK-1 focused on colour classification of visual landmarks and on mechanisms to attend to different candidate landmark locations in an image.

Detecting Landmarks and Objects Using Colour Visually searching for objects requires scanning the environment or checking expected locations with a camera. When searching for a landmark the robot can predict where to point the camera as it knows its own approximate location on the map and the coordinates of the landmark. Uncertainty in the robot's position requires selecting a wide field of view for the camera. An attention mechanism that selects potentially "interesting" locations in an image or environment significantly speeds up and simplifies the search. Features such as intensity, colour, high contrast, motion and presence of significant edges are often used to focus attention. Once candidate locations have been selected, each of them is inspected closely to verify the presence of the target object.

Colour can be used to identify possible candidates in an image. The ARK colour classification scheme consists of an off-line training phase and an on-line classification of pixels on a real-time image processor[7]. Colour information is used for pixel wise classification of images and assigning pixels to possible target candidates or background classes. Real-time performance is achieved by creating look up tables (LUTs) during the training phase and using fast indexing during the on-line classification.

In the off-line training phase, the training set consist of images with objects of interest in their natural environment and under different illuminations. Each of the pixels in the training set is described by its hue, saturation and intensity. These are obtained from the measured RGB values. The training data is first re-sampled to create reduced images by factors of 16 and 4 respectively. Then the K-means clustering algorithm is applied to the smallest image first and starts from random seed points. The cluster centers obtained from the smallest images are used to seed the clusters at higher resolutions. The process lasts several minutes, and achieves a good partitioning of the data. After clustering the user assigns individual clusters to classes that correspond to objects of interest.

Given a test image, a classification algorithm is used to process all the pixels in the test image, filling all the cells of the resulting LUT. This operation takes several minutes on a standard workstation. A representation for each class is stored and used for classification. Assuming multivariate normal distributions of clusters in the colour space and equal a priori probabilities for each cluster, the Bayes discriminant function can be used [7]:

$$g^{(k)}(x) = (x - m^{(k)})^T cov_k^{-1} (x - m^{(k)}) - log|cov_k|$$

Where $m^{(k)}$ is the centroid of the k-th cluster and cov_k is its covariance matrix. The classification of the image is performed by calculating the discriminant function for every pixel described by vector x and every cluster k. The class assigned to the pixel is the one that minimizes the value of the discriminant function. The computational complexity of this technique depends on the number of clusters and the resolution of the LUT. Classification of every pixel in the image is therefore a computationally expensive task. Modern image processing systems are often equipped with large LUTs that allow for real-time processing of every pixel. Combination of multiple data streams, for example RGB, into one channel enables us to index into the LUT and achieve the real-time performance of an arbitrary (non-linear) conversion. The nature of this conversion is determined by the contents of the LUT. The problem is how to create a LUT that will effectively capture the important variability of the data.

Resolution of the feature space can reach 2^{24} (3 x 8 bit colour bands) for standard colour cameras. Often it is sufficient to operate on smaller arrays. There are hardware limitations as well, for example, the Datacube MV20 advanced processor used in the project has an LUT with a maximum of 64K entries. The contents of LUTs are often determined by manual selection. A more systematic approach uses training by showing examples and manually delineating the objects of interest. Cells in colour space, corresponding to the feature combinations present in the training set, are assigned to appropriate classes. For low resolution of the feature space (200 cells) such a technique is sufficient, as camera noise and blur create dense clusters[21]. For high resolution LUTs containing, for example 64K cells, this approach is not reliable as insufficient training data creates “holes” in the feature space. Such holes cause misclassification of the data. Various heuristic techniques of filling the space have been used to bridge the gaps [17].

(a) Office Scene

(b) Detected objects

Figure 5: An office scene with coloured objects (luminance is shown only).

The training phase (clustering and creation of the LUT) is implemented on a Unix host. The real-time colour classification is implemented on a MaxVideo 20 image processing system. The classifier is trained to detect red and green circular plates similar to the ones displayed on the wall in the scene shown in Figure 5(a). Figure 5(b) shows the results of pixelwise classification, filtering and reconstruction of large blobs representing red and green classes. The results of this processing are not perfect - both red plates have been detected but among the four green candidates only one corresponds to the target object. Also, detection of individual plates is not perfect as regions in the shade or reflecting light are misclassified. Different techniques could be used to decide whether the detected blobs correspond to valid objects or not. At this resolution, however, it might be difficult to decide if the shape deformations are caused by noise, particularly if the sensor is positioned at a difficult viewing angle. It is much better to simply point the robotic head at every candidate in turn with a narrower setting of the zoom lens and then acquire and process a new set of images.

Each detected candidate is described by a set of parameters that define its position in the image, size and location of its bounding window. The new orientation of the head is calculated from a kinematic model of the head that includes the pan, tilt and the initial size of the field of view. The new setting for zoom is selected so that the blob of interest is not only fully included in the new view, but also dominates the field of view.

Correlation-based landmark detection The landmark-finding module used in ARK-2 is based on performing a multi-resolution normalized correlation between a query image (from the robot's current position) and a *reprojection* of the stored "3D grey-level surface"

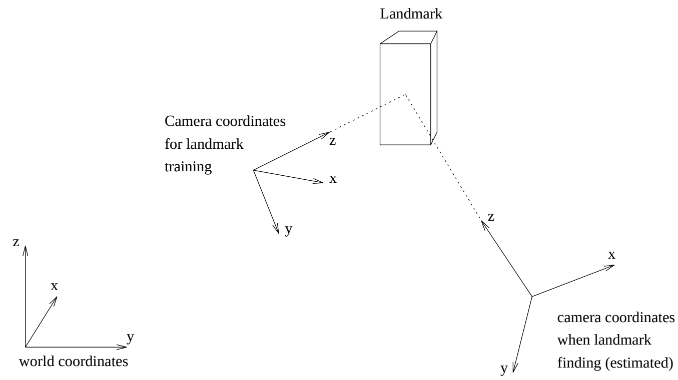


Figure 6: Landmark reprojection geometry

representing the landmark. The grey-level surface is a resolution pyramid consisting of registered grey-level images of the landmark at various resolutions, and the estimated depth of each pixel in 3-D as seen from the training position. The idea is to use the robot's estimate of where it is, plus knowledge of the viewpoint from which the landmark was learned, to make an accurate enough prediction of the appearance of the landmark in order to match it successfully in the query image.

Figure 6 illustrates the geometry used for the landmark reprojection. When a landmark is learned, a coarse range scan is done of the area covered by the stored grey-level image. The scan may be a simple grid of points, or an adaptive scan that focuses sampling on areas of non-linear variation in depth with image position [28]. Depth values are interpolated for each pixel in the image, so that each pixel may be assigned a position in 3-space. The positions are stored in the coordinate system of the camera at the position at which the landmark is learned.

When a landmark is to be found for a position correction, the approximate position of the robot, as given by the dead-reckoning system, is used to determine the position of each pixel in the current camera image coordinate system. These grey-level values at selected image locations are used in an interpolation to compute the predicted appearance of the landmark from the estimated robot position. The multi-resolution normalized correlation between the central region of the reprojected landmark image and the query image works extremely well, provided that the appearance of the landmark is unique, at all resolutions, in the field of view.

The use of multi-resolution matching achieves two objectives: First, it reduces computation time dramatically by allowing operation on several small images in place of a single large image. Second, it allows computation of a figure of merit based on the fact that good matches at the coarser level should result in a match at the centre of the finer level, for each pair of levels.

This multi-resolution technique performs sufficiently well for the robot to navigate the AECL test environment successfully. It still requires care, however, to choose landmarks that will work consistently well. One of the key things to remember is that the landmark must appear distinctive at not only the highest camera resolution, but also at lower resolutions. As well, the landmark should have simple structure in depth (smooth surfaces are good) if reprojection from different viewpoints is desired. This eliminates holes in the data as occlusions of surfaces change.

5 Sensing for safety

In addition to dealing with pose estimation and correction, a mobile robot requires sensors to deal with maintaining the safety of the robot. The various ARK robots have used a number of visual and non-visual sensors; sonar, IR and bumpers to form an extended virtual bumper around the robot.

5.1 Floor anomaly detection

The floor of an industrial environment can be very complex. The AECL bay, for example, contains drainage ditches (which can be open), cables, ducts, etc., which are temporally varying structures which can prevent the safe passage of the robot. Note that unlike wall structure in a corridor environment which will typically be sensed by touch sensors if the robot approaches the obstacle too closely, the drainage ditches in the AECL bay could simply cause the robot to fall into them and tip over, resulting in serious damage to the robot. Before moving the robot onto a particular piece of the floor it is important to insure that the floor is traversable. Three different approaches to floor anomaly detection were considered in the ARK project.

5.1.1 Floor anomaly detection using combined vision-range measurements

One obvious technique for determining that the floor in front of the robot is passable would be to probe the ground in front of the robot with the laser scanner. Given the relatively slow speed of the laser scanner, this approach would seriously degrade the performance of the robot if many laser probes are necessary. One approach to reducing the number of probes necessary to survey the scene in front of the robot is to use the vision sensor to determine an initial segmentation of the space in front of the robot and then to make selective range measurement in each region to verify the floor in front of the robot. This assumes that depth discontinuities coincide with boundaries of detected regions. To satisfy this assumption, initial segmentation parameters are tuned so as to create an over- rather than under-segmented representation of the intensity image. Thus the need for region splitting is avoided. The initial segmentation creates an image tessellated into primary regions of homogeneous image properties (intensity, colour, etc.). The segmentation method adopted for the project consists of smoothing, morphological edge detection and the watershed transform [26]. The segmented image is represented as an adjacency graph that includes region descriptors derived from the original image and their topology (adjacency of regions and boundaries, connectivity of curves, etc.).

Five range measurements were chosen per region. For each measurement, the elevation, azimuth and distance was recorded. After finding the target positions in the World Coordinate System (obtained by calibrating Laser Eye in advance), each region was approximated as a plane using a least squared error criterion.

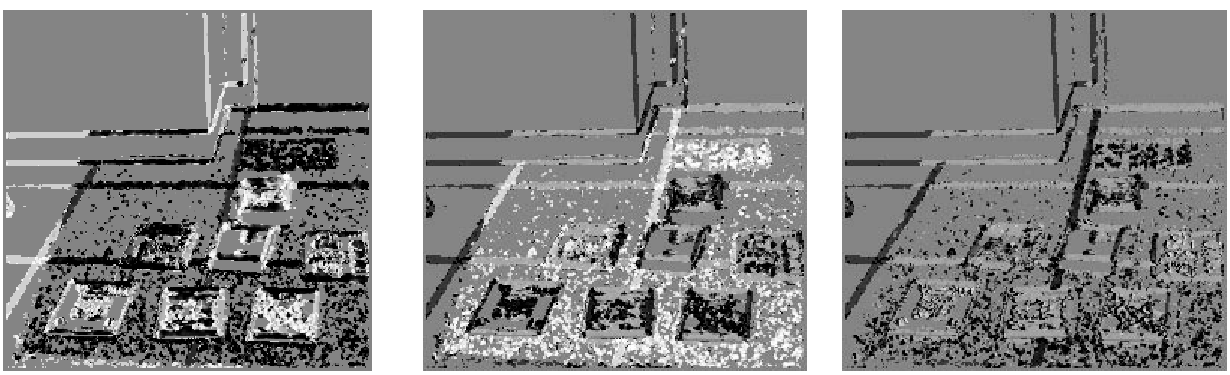
For floor anomaly detection, calibration consists of two steps. The first step involves the measurement of the intrinsic and extrinsic camera and head parameters. The second step involves the measurement of the reference floor plane. During normal operation, the robot directs the camera in the direction of travel, acquires an image and creates a representation of the scene. The head sweeps the scene in front of the robot by directing the Laser Eye at selected regions. Each region is verified whether it belongs to the floor or not. The algorithm continues to build the maximum floor coverage. The verification process uses the distance from the reference floor plane to accept or reject the region.

5.1.2 Floor anomaly detection using stereo vision

Another approach is to use stereo vision to verify that the floor in front of the robot is solid[10]. In a typical stereo vision application, objects in one camera are matched with objects in the other and these correspondences coupled with the known geometry can be used to identify the three dimensional location of structure in the environment. Perhaps the most difficult task in stereopsis is the determination of the correspondence of features in one camera with features in the other. For a Floor Anomaly Detector (FAD), however, it is not necessary to determine the correspondences for arbitrary scene structure. Rather it is only necessary to determine correspondences for structure that lies near a particular 3D plane (the floor). If the cameras are modeled as pinhole cameras then it is possible to warp one of the images so that the floor has zero disparity (see [5]) which simplifies the matching process considerably.

Figure 5.1.2a shows a sample stereo pair of the floor cluttered with obstacles. Figure 5.1.2b shows the recovered obstacles which have been classified using a robust statistical technique based on mixture models[12] to group the raw disparity measurements into three pools; pixels which are consistent with the floor plane model, pixels which are near the floor plane (anomalies), and pixels which could not be classified. The technique is fairly straightforward, reasonably efficient and quite robust provided that sufficient image structure exists on the floor. Unfortunately many floor surfaces are reasonably featureless

(a) Stereo pair of the floor



(b) Measurements consistent with obstacles (left), the floor (centre), and outliers (right). Intensity encodes class probability with white representing high probability.

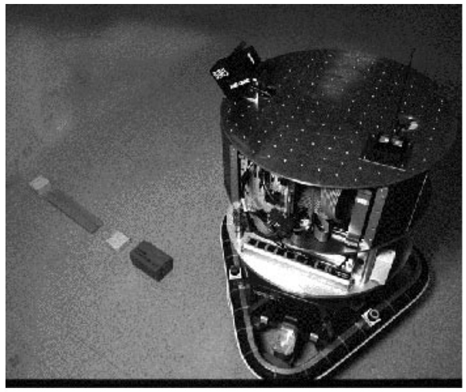


Figure 7: BIRIS-based FAD

and do not provide a rich surface texture for stereo matching. One possible mechanism for overcoming this problem is to project some random texture onto the floor to break up this camouflage.

5.1.3 Floor anomaly detection using laser stripes

A third approach to FAD is based upon the use of a laser stripe device using the BIRIS sensor developed by the National Research Council (NRC) in Ottawa [2]. The basic optical principle of this method is a combination of optical triangulation and of the use of a video camera with a double aperture mask in the iris plane of the camera lens (hence BI-IRIS). A laser stripe is projected on the floor in front of the robot and a BIRIS sensor is used to recover the position of the projection and hence the floor depth. If the floor is flat, then the floor depth should remain constant. Any variation in the floor plane can be detected in a straightforward manner. A mobile robot equipped with FAD can avoid obstacles in real time at speeds of up to 0.5 m/s. Figure 7 shows a mobile robot with the FAD mounted on it.

This particular approach has two problems from an industrial standpoint. The first is that the laser used is not eye-safe, and thus there are safety concerns, especially if there are reflective materials – such as pools of liquid – on the floor. The second is that this technique relies on a line process which means that the safe floor region is that region that has been swept out by the motion of the line, and the robot must be controlled to only move through that region which has been “cleared”.

5.2 Segmenting space

Although the most common use for Laser Eye is in performing measurements for odometry correction, it may also be necessary to sense unknown or partially known areas in order to

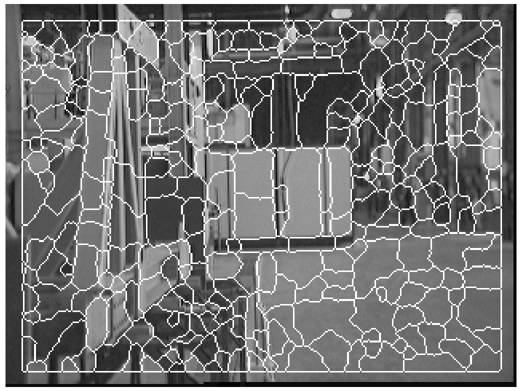


Figure 8: Segmentation of the AECL bay.

determine if they are passable. This would occur in the field when exploring the environment after some disaster had occurred. Thus one additional task to which the ARK active vision sensor has been put is to segment the volume of space in front of the robot in order to obtain a depth map which can be used to determine if the way in front of the robot is possible.

One possible mechanism for determining this depth map would be to sample densely the volume of interest. Given the relatively slow performance of the laser system, for real-time operation of the robot it is important to minimize the number of measurements. Fortunately, visual image data can be used to plan where to point the range-finder [7, 8, 6].

Let us assume that nearly all significant depth discontinuities in the scene coincide with the boundaries of detected regions. As in floor anomaly detection using combined vision-range measurements, this assumption requires that the initial segmentation creates an over-segmented rather than under-segmented representation of the image. The under-segmentation can cause potential problems as it requires additional depth measurements to split the region along a depth discontinuity. The size of the regions should not be too small as it is difficult to obtain reliable distance measurements for small regions due to the finite size of the laser spot and accuracy of the robotic head.

The initial segmentation creates an image tessellated into primary regions of homogeneous image properties (intensity, colour, etc.). The segmentation method adopted for the project consists of smoothing, morphological edge detection and the watershed transform (see [7]). Large numbers of closed regions of similar image properties are created as a result as shown in Figure 8.

For the scene shown in Figure 8 the initial segmentation created almost two hundred primary regions. Assuming the simple model with one range measurement per region, creation of the complete range map requires almost 200 range measurements. By applying the above technique we have been able to reduce the number of range measurements

Figure 9: Selection process for uniformly biased model

required to create the dense range map from 64 K samples (sampling every pixel in a 256x256 grid) to a much more manageable number of 200 to 1000 samples (200 regions x 1...5 targets per region). This has been achieved if the initial over-segmentation of the image identified intensity discontinuities and that they account for nearly all the depth discontinuities. For the mobile robot, operating in real-time, this may still be too slow. If we look at the intensity image ourselves, it seems that a few range measurements, taken in the "right" directions, could provide the essential information for a specific task. We decided to look to models of human attention for inspiration.

The attention scheme used here depends on three components [19]: (i) a priori information, (ii) selection of salient features, and (iii) a given task and previous results of attentive processing. The *a priori* information is encoded as a function biased to look at specific parts of the image. This function represents preferred behaviour (directional sensitivity) of the system, for example, data in the centre or below the horizon might be more important than at the periphery of the camera image. Representing the segmented image data as a graph allows easy access to underlying regions and boundaries in the graph and for access to adjacent ones. The regions are described by features such as intensity, colour, texture descriptors, and their size and shape. The boundaries between adjacent regions are described by their size, shape, orientation and contrast between regions on both sides. Detection of winners, in the "Winner Take All" scheme [25], uses a combination of these features and is biased by the specific task performed by the robot.

For example, looking for a passage might involve searching for a dark region in the image. Depth discontinuities are likely to occur at boundaries between contrasting regions. If the task is to provide a qualitative range map, then selecting large regions first will enable faster coverage of the image by range data. Results of previous range measurements can influence the selection of the next target. This selection is task dependent. For example, when searching for an obstacle, if a depth discontinuity is detected, then the next ranging

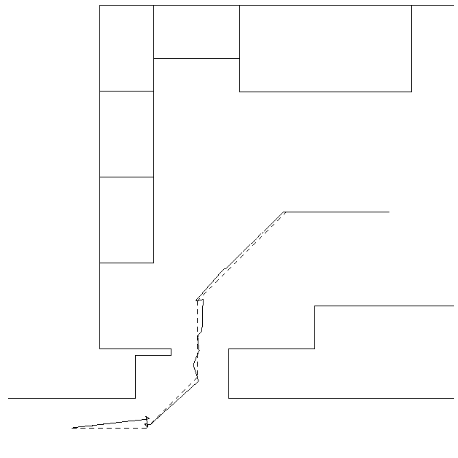


Figure 10: Reactive robot control. The robot moves from the upper left to lower right in the figure. The commanded path is given by a dotted line while the actual path followed is drawn as a solid line. Modelled environmental obstacles are also shown.

operations should concentrate on recovering the full extent of the closer object and not the distant one. If such a discontinuity is detected while searching for a passage then the successive ranging operations should concentrate on objects further away - the opposite strategy.

Figure 9 shows the attended receptive fields and the path of 10 saccadic movements between regions of high intensity. The initial bias is uniform and contributions from all receptive cells (pixels) are treated equally and, as the result, large bright regions are attended first. Edges of high contrast are likely locations for depth discontinuities. Boundaries between regions now serve as salient features. Pointing the range-finder at a boundary is not practical so two regions on both sides are selected for attention.

6 Navigation and control

The ARK control system consists of two levels: a high level and a low level reactive system. The high level is responsible for planning robot actions, global path planning, selecting landmarks for sighting and interactions with the user. The low level, reactive component of the control system, uses the on board obstacle avoidance system of the platform to detect obstacles and to navigate around them (see Figure 10).

The path planner assumes that the low level reactive control structure will safely execute segments of the plan in the presence of unmodelled or unexpected obstacles. By breaking the path planning process into a GOFAIR (Good Old Fashioned AI and Robotics) task which can be processed using classical AI tools, and a real time reactive process which can be processed using a real time safety critical system implemented as a subsumption architecture, ARK takes advantage of the best of both paradigms.

Although this hierarchical control model is common over all three ARK robots, each of the robots have different levels of autonomy built into each of the layers. For example, ARK-1's reactive layer is trivial in that it simply halts the robot should something enter within a pre-defined safety radius. At the other extreme, while ARK-lite implemented a sophisticated reactive control structure which is described below. ARK-2 relies on a modified version of the ARK-1 control architecture but includes a planning module to allow the robot to navigate around small unexpected obstacles.

6.1 Map and Path Planning

At the high level, the ARK robots represent the world as a simple occupancy grid describing each 10-centimeter square of the floor as either empty or occupied. Planning is done by computing a "potential field" for each empty cell in the grid, whose value is a function of the proximity of obstacle cells. The "optimal" path between two operator-specified way points is considered to be the path minimizing the path integral of the potential field. This is a traditional approach to path planning.

The approach of minimizing the path integral of the potential is effective because it balances length of path against the difficulty of the path, as expressed by the potential field. Obviously, other terms could be included in the field to account for things such as the visibility of landmarks, or other robot hazards. And paths could be computed which take these events into account (see [9] for example).

Various classical path planning techniques have been used to plan paths through this discretized representation of the robots workspace. ARK-1 and ARK-lite use a configuration-space representation in which the $10cm \times 10cm$ cells are further divided into discretized orientations, while ARK-2 only encodes the position. Classical path-planning using either A* or uninformed graph search was found to be effective for ARK-1 and ARK-lite, while ARK-2's path planning is computed using a modified Dijkstra's algorithm on a discrete mesh of possible robot positions. Partial paths in the mesh are only kept by the algorithm if they have length less than a constant multiple of the straight-line distance from the start to the end of the partial path. Although this may in theory cause the only possible path between two points to be missed by the search, we have found that in practice, substantial pruning of the space of paths examined in finding the best path is possible without affecting the result of the algorithm. We achieve $O(n \log n)$ complexity for the algorithm, for n the number of mesh points, as follows. The set of nearest mesh points not yet expanded by the algorithm is maintained in a heap ordered by distance from the starting point. The

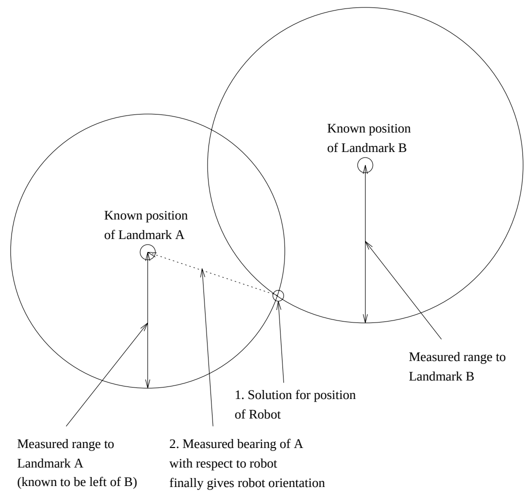


Figure 11: ARK-2 navigation geometry

position of each mesh point in the path search is maintained in a data structure, to allow $\log(n)$ update of the heap position of each point as the length of the shortest path through the point is updated as new paths are explored. A shortcoming of this approach is that the number of mesh points for path planning grows as the square of the inverse of the desired mesh point spacing. As a result, path planning can be slow for long paths. A simple and attractive approach to reducing path planning time is to use precomputed “highways” for the robot down main corridors, with path planning restricted to the portions of travel leading to and away from the nearest highway.

6.2 Position Estimation and Navigation

The ARK robot maintains its estimate of where it is located and which way it is heading in much the same way as a sailboat performs coastal navigation. Periodic position fixes are done based on mapped landmarks in the local area, with dead-reckoning in between position fixes to estimate the current position at all times.

As can be seen from Figure 11, the range to two known points in a plane and the robot-relative pan angle to one of the points is sufficient to determine the position and orientation of the robot on the same plane. There are in fact two solutions for position, but no ambiguity as long as it is known which landmark is to the right from the robot's viewpoint.

The range along the floor to each landmark may be obtained by one of two means. Using the laser range finder on the Pan-Tilt Unit, the range to the landmark may be obtained directly. The elevation of the landmark may then be used to determine the projection of its range onto the floor. If the landmark does not lie in the plane of the range finder then an estimate of range is available from the elevation and a consistency check can be performed. Since it is possible to predict the pan angle separating the two landmarks as seen from the robot's corrected position, there is a consistency check on the position correction available via the comparison of this expected separation with the actual measured separation. Other over constrained sensor-based methods have also been considered for the robots, see in particular Lu and Milios [14, 15].

6.3 Reactive Control

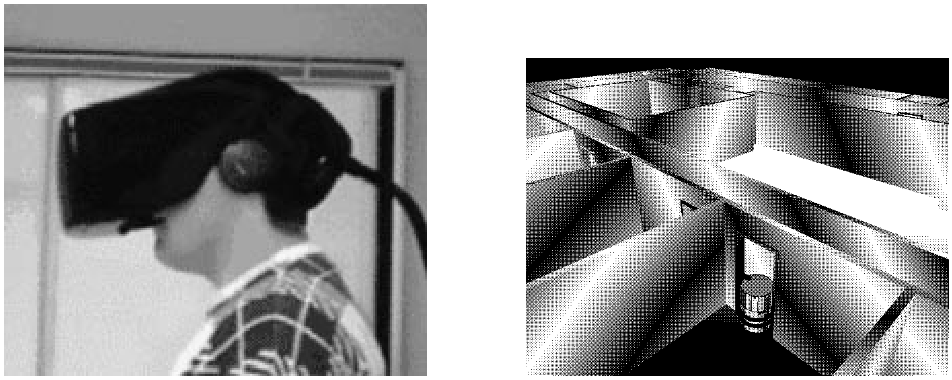
The high level planner communicates with the reactive subsystem through a very simple set of operations that assumes the reactive phase of the planner will operate autonomously and asynchronously attempting to achieve the current subgoal. ARK-1 and ARK-2 assume a "stop and shoot" model of low-level control. When a local portion of the path cannot be executed due to an unmodelled or unexpected obstacle, the robot stops and performs various sensing tasks to determine a path around the obstacle. ARK-2 relies on a more reactive low-level control mechanism [20] which is based on the subsumption approach described by Brooks[3].

On ARK-lite, the robot is guided by a set of behaviours that operate in parallel. Each behaviour maps a sensory reading from the robot's environment into an external action of the robot. Conflicting behaviours are arbitrated based on an absolute prioritization of behaviours. There are three basic behaviours that control the robot: move, avoid, and escape. The Avoid behaviour watches for an obstacle detected by the front sensing sonar. If an object appears the avoid behaviour stops the robot, and turns it to a new direction so that the robot will not collide with the obstacle. The escape behaviour watches for an obstacle directly in front of the robot, in which case, it causes the robot to back-up and then to turn to a new direction. The escape behaviour helps to get out of certain deadlocks that may occur with the avoid behaviour when the robot gets stuck in a corner. The move behaviour steers the robot towards a precomputed goal position.

6.4 A 3D immersive display for robotic control

In an operational setting, the ARK robot requires an operator to provide high-level mission commands. These high-level commands can be provided via a $2\frac{1}{2}$D map-based user interface as well as through an immersive 3D interface. The 3D interface provides the operator with a virtual reality-like control interface. It allows the operator to move through a simulation of the robot's environment, to examine the environment through an immersive display, and provides access to high-level mission commands in a more informative and natural way than is possible with the standard $2\frac{1}{2}$D map-based user interface.

The standard user interface for the ARK robots is based on a $2\frac{1}{2}$D map similar to the ARK-lite user interface shown in Figure 12a. The map is primarily 2D but does contain



(a) Operator (b) Operator's view

Figure 12: Immersive user interface

some height information. Through this $2\frac{1}{2}$D interface an operator can command the robot to plan a path from its current location to a specified goal location, execute a pre-planned path, and perform other high level tasks. The user indicates operations using a set of buttons and indicates locations in the robot's environment by pointing with a mouse on the displayed map. Although this user interface was found to be sufficient for a number of tasks, it was found that the operator experienced difficulties when trying to make fine motions of the robot and had difficulty visualizing the robot's operating environment.

Driving a remote device with a joystick or some other similar input mechanism can be very difficult. Unless extensive sensor measurements of the environment are available and presented to the operator in an effective and timely fashion, it can be very easy for the operator to become disoriented with respect to the remote environment, which can lead to operator error.

In order to construct an advanced teleoperational interface for a mobile robot, it is necessary that the interface be consistent, integrated, and natural to use. Mechanisms which rely on a large bank of monitors, with complex user interactions cannot be expected to provide a natural input mechanism. One technology which can be exploited to provide a more natural interaction mechanism is an immersive display or virtual reality technology [1].

For an immersive display to provide an effective mechanism for control of a mobile robot, the interface must do at least two things; it must provide the operator with a useful representation of the robot's operational environment, and it must provide suitable interaction mechanisms for robotic or teleoperational control. For the immersive environment to provide a useful representation of the robot's operational environment, the operator should be able to view, and navigate through, the environment. For the entire interface to provide interaction, some mechanism for operator input beyond that required for the immersive display must be provided.

The ARK-lite immersive interface is based around a head mounted display (HMD) and a six degree of freedom joystick. Video is displayed on a Liquid Image HMD which also provides stereo sound to the operator. Six degree of freedom (DOF) head tracking is accomplished via a Flock of Birds head tracker. The operator is also equipped with a six DOF Cyberman three button joystick to provide additional input control. Video is generated by an SGI Indigo^2 workstation with the Extreme graphics option.

A fundamental question in the design of an immersive interface for a mobile robot is how to manage the display of both the immersive visual display as well as any visual tokens which must be displayed as part of the interaction mechanism. The display portion of a head mounted display can be considered as a simple flat display surface, but interaction mechanisms which are appropriate on "flat" monitors are unlikely to be well suited for head mounted displays.

Although the display surface of the Liquid Image HMD does subtend a relatively large visual angle, its actual display surface is quite small. With a visual field 640x480 pixels in size, there is not much physical screen real estate to reserve for any graphics required for interaction. In addition, due to the magnification optics built into the HMD, it is only possible to read the center of the screen without strain.

Given the need for graphical displays not related to the immersive display, limited screen real estate, and the fact that the best view is in the center of the screen, a user interface is required that is in some sense foveal. Thus the ARK-lite immersive display introduces a *fish bowl* metaphor for the control and manipulation of graphical objects.

The fish bowl metaphor is an extension of the desk-top metaphor common in 2D graphical user interfaces. Imagine being a fish in a fish bowl. Looking out through the walls of the fish bowl you can view the environment within which your bowl sits. The external world outside the fish bowl projects onto the bowl's exterior surface. The interior surface of the bowl completely surrounds the operator providing 360^o of desk-top surface. Semi-transparent and opaque 2D graphical objects can be placed on the surface of the bowl. Interaction mechanisms are provided so that the operator can:

- Translate the operator and the bowl through the external environment.
- Rotate inside the bowl to view out through different portions of the bowl. This is known as the pan model of operation.
- Rotate with the bowl so that the objects on the surface of the bowl obscure different regions of the external environment. This is known as the fixed model of operation.
- Select objects on the surface of the bowl and

- Move them to other locations on the surface of the bowl, including placing them on top of other objects on the surface of the bowl.
- Dispose of them.
- Resize them.

As the operator's field of view is limited, only a portion of the fish bowl is visible at any one time.

In order to select different graphical objects on the fish bowl for input focus, the operator simply rotates until that object is in the center of view. i.e., the operator simply looks straight at the object of interest. A cross-hair is always displayed in the center of the display to aid the operator determine which interaction object is currently receiving input focus.

7 Discussion

In order to effectively deploy a mobile robot in an industrial environment it must be safe, reliable, and easy to use in addition to performing some task that is difficult, disagreeable, or expensive for a human to perform. Conducting survey/inspections within an industrial environment such as a nuclear or chemical plant environment meets the task requirements. The task is repetitive and when an anomalous situation is detected within the plant, the task becomes disagreeable and can be highly dangerous. From an economic point of view it is perhaps an idea task for a mobile robot.

The task introduces a number of technical problems which must be addressed if a mobile robot can be applied to the task: The robot must be able to perform point to point navigation with respect to a global environmental map. The robot must be safe in that it can successfully detect and react to unexpected or unmodelled obstacles to its motion. The robot must provide an effective mechanism for a trained operator to interact with the robot. The ARK robots have developed effective solutions to these tasks.

Fundamentally, the ARK robots rely on *Laser-Eye*, a combined vision and range sensor, to navigate through the industrial environment. Laser Eye is unique as it operates at the large distances typical in industrial settings. This sensor allows the robot to detect landmarks, search for objects and possible paths through its environments. Combined with a set of pre-mapped visual landmarks, this sensor "solves" the problem of global navigation within an industrial environment.

By endowing the robot with other sensor modalities including laser line stripers, stereo cameras, sonar, IR and bumpers, the robot can obtain sufficient local environmental information to deal with unmodelled and unexpcted obstacles to its motion including failures in the underlying floor itself. A number of algorithms were also developed to explore the application of Laser-Eye to identifying passageways within the environment and to determining the structure of objects in the environment.

The ARK robots rely on a layered control architecture in which lower levels essentially transduce sensor measurements into motion commands in order to provide a fast response to unexpected obstacles. Layered above this safety control system there exists a navigational unit which provides reliable point-to-point navigation within the robot's environment. Breaking the vehicle control into these two levels allows the continuous nature of the time- and safety-critical system to operate in conjunction with the discrete higher-level navigation functions.

Finally, the ARK project developed a novel immersive user interface system for mobile robots to complement the classical "point and go" model of robot control.

As delivered, ARK-2 meets its task requirements within a modern industrial environment. It performs point-to-point motion within its environment while avoiding and dealing with unexpected obstacles. It is also capable of performing related tasks such as measuring iso-contours of events such as temperature and gas concentations. In addition to meeting these technical goals, research undertaken as part of the project has lead to advances in general mobile robot system (eg., [24]), image understanding (eg., [16, 27, 11]), system control (eg., [23]), and immersive user interfaces (eg., [1]).

Acknowledgments

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., Industry Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro, and Atomic Energy of Canada Limited. This project would not have been possible without the dedicated efforts of N. Aucoin, J. Bruce, T. Campbell, B. Down, F. Lu, B. Majarais, M. Robinson, O. Sanbekkhaug, R. R. Service, D. Terzopoulos, K. Tran, and V. Wu.

References

[1] N. Aucoin, O. Sandbekkhaug, and M. Jenkin. An immersive 3d user interface for mobile robot control. In *IASTED Int. Conf. on Applications of Control and Robotics*, pages 1–4, Orlando, Florida, 1996.

[2] F. Blais, M. Rioux, and J. Domey. Compact three-dimensional camera for robot and vehicle guidance. *Opt. Lasers Eng.*, 10:227–239, 1989.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. of Robotics and Automation*, 2:14–23, 1986.

[4] B. Down, E. Milios, M. Jenkin, P. Jasiobedzki, T. Capbell, and J. Tsotsos. LASER EYE: Imaging and ranging apparatus and aiming method. Canadian Patent 2,105,501.

[5] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, MA, 1993.

[6] P. Jasiobedzki. Active image segmentation using a camera and a range-finder. In *Application of Artificial Intelligence 1993: Machine Vision and Robotics, Vol. 1964*, pages 92–99. SPIE-The International Society for Optical Engineering, 1993.

[7] P. Jasiobedzki, B. Down, and V. Wu. Active object detection using colour. In C. Archibald and P. Kwok, editors, *Research in Computer and Robot Vision*, pages 37–52. World Scientific Press, Singapore, 1995.

[8] P. Jasiobedzki, M. Jenkin, E. Milios, B. Down, J. Tsotsos, and T. Campbell. Laser Eye - a new 3D sensor for active vision. In *SPIE Intelligent Robotics and Computer Vision: Sensor Fusion VI*, pages 316–321, Boston, MA, 1993.

[9] M. Jenkin, E. Milios, P. Jasiobedzki, N. Bains, and K. Tran. Global navigation for ARK. In *Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, Yokohama, Japan, 1993.

[10] M. R. M. Jenkin and A. Jepson. Detecting floor anomalies. In *Proc. British Machine Vision Conference*, pages 731–740, 1994.

[11] A. Jepson and M. Black. Mixture models for optical flow computation. In Ingemar Cox, Pierre Hansen, , and Bela Julesz, editors, *Proceedings of the DIMACS Workshop on Partitioning Data Sets: With Applications to Psychology, Vision and Target Tracking*, pages 271–286, Providence, RI, 1995. AMS Pub.

[12] A. D. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proc. IEEE CVPR*, pages 760–761, New York, NY, 1993.

[13] S. Kamewaka and S. Uemura. A magnetic guidance method for automated guided vehicles. *IEEE Trans. on Magnetics*, 23(5), 1987.

[14] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, to appear.

[15] F. Lu and E. Milios. "globally consistent range scan alignment for environment mapping". *Autonomous Robots*, 4(4):333–349, 1997.

[16] W.J. MacLean, A. Jepson, and R. Frecker. Recovery of egomotion and segmentation of independent object motion using the em-algorithm. In *Proc. of British Machine Vision Conf.*, pages 175–184, 1994.

[17] R. Massen and C. Volk. Real-time colour classification for preprocessing photogrammetry images. In *Proc. of SPIE: Close-Range Photogrammetry Meets Machine Vision, vol. 1395*, pages 283–290. SPIE, 1990.

[18] B. Nickerson, M. Jenkin, E. Milios, B. Down, P. Jasiobedzki, J. Tsotsos, N. Bains, and K. Tran. ARK - autonomous navigation of a mobile robot in a known environment. In *Proc. of International Conference on Intelligent Autonomous Systems: IAS-3*, pages 288 – 296, Pittsburgh, PA, 1993.

[19] Piotr. a paper on attentive processing using lasereye. In ???, editor, *????*, page ??? ???, ???, 1994.

[20] M. Robinson and M. Jenkin. Reactive control of a mobile robot. In C. Archibald and P. Kwok, editors, *Research in Computer and Robot Vision*, pages 55–70. World Scientific Press, Singapore, 1995.

[21] M. Swain and D. Ballard. Color indexing. *Int. Journal on Computer Vision*, 7(1):11–32, 1993.

[22] TRC. Beacon Navigation System. Product Literature, Transitions Research Corporation, 1994.

[23] J. K. Tsotsos. Intelligent control for perceptually attentive agents: The s* proposal. *Robotics and Autonomous Systems*, (in press).

[24] J. K. Tsotsos, G. Verghese, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflo, S. Stevenson, M. Black, D. Metaxas, S. Culhance, Y. Ye, and R. Mann. PLAYBOT: A Visually-Guided Robot for Physically Disabled Childred. *Image and Vision Computing*, to appear.

[25] J.K. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78(1-2):507 – 547, 1995.

[26] L. Vincent. Watersheds in digital spaces: an efficient algorithm based on immersion simulation. *IEEE PAMI*, 13(6):583–598, 1991.

[27] Z. Wang and A. Jepson. A new closed-form solution for absolute orientation. In *Proc. of IEEE Conf on Comp. Vis. and Pat. Rec.*, pages 129–134, Seattle, WA, 1994.

[28] D. Wilkes. something about adaptive depth scan that focuses sampling on areas of non-linear variation. *????*, ????:???, ??? ???

[29] D. Wilkes, G. Dudek, M. Jenkin, and E. Milios. Multi-transducer sonar interpretation. In *IEEE Int. Conf. on Robotics and Automation*, pages 392–397, Atlanda, GA, 1993.