ELSEVIER

Robotics and Autonomous Systems 25 (1998) 83-104

Robotics and Autonomous Systems

The ARK project: Autonomous mobile robots for known industrial environments

S.B. Nickerson [a], P. Jasiobedzki [b], D. Wilkes [a,c], M. Jenkin [c,*], E. Milios [c], J. Tsotsos [b], A. Jepson [b], O.N. Bains [d]

[a] *Ontario Hydro Technologies, Toronto, Ontario, Canada*
[b] *Department of Computer Science, University of Toronto, Ontario, Canada*
[c] *Department of Computer Science, York University, North York, Ontario, Canada M3J 1P3*
[d] *Atomic Energy of Canada Ltd., Mississauga, Ontario, Canada*

Received 10 October 1997; received in revised form 28 May 1998

Abstract

The ARK mobile robot project has designed and implemented a series of mobile robots capable of navigating within industrial environments without relying on artificial landmarks or beacons. The ARK robots employ a novel sensor, *Laser Eye,* that combines vision and laser ranging to efficiently locate the robot in a map of its environment. *Laser Eye* allows self-location of the robot in both walled and open areas. Navigation in walled areas is carried out by matching 2D laser range scans, while navigation in open areas is carried out by visually detecting landmarks and measuring their azimuth, elevation and range with respect to the robot. In addition to solving the core tasks of pose estimation and navigation, the ARK robots address the tasks of sensing for safety and operator interaction.

Keywords: Mobile robots; Sensing; Industrial robots

1. Introduction

There are many types of industrial operations and environments for which mobile robots can be used to reduce human exposure to 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 envircmments; spill cleanup, and the inspection and carrying out of repairs in the radioactive areas of nuclear plants. The latter application leads to increased safety by reducing the potential radiation dose to workers. It is this industrial survey and inspection task that is addressed by the ARK (Autonomous Robot for a Known Environment) project.

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 significant application area for mobile robots in industrial environments is to provide independent verification

* Corresponding author.

0921-8890/98/$ – see front matter
PII: S0921-8890(98)00032-3

of existing instrumentation. Another application area involves performing teleoperated repairs, where it is desirable for the robot to arrive autonomously at its destination before the teleoperated repair starts. For a mobile robot to be able to perform such tasks, it must be possible to direct the robot to a specific location described in a global metric coordinate system: it is thus essential that the ARK robot knows 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 Atomic Energy of Canada Limited (AECL) in Mississauga, Ontario. This open area covers approximately 50000 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 an 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, with forklift trucks and floor cleaning machines in operation. This type of an environment presents many difficulties for a mobile robot including: the lack of vertical flat walls; large open spaces (the main isle is 400 ft long) as well as small cramped spaces; high ceilings (50ft); large windows near the ceiling resulting in time dependent and weather dependent lighting conditions; a large variation in light intensity; 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 [17]. Fig. 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 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.

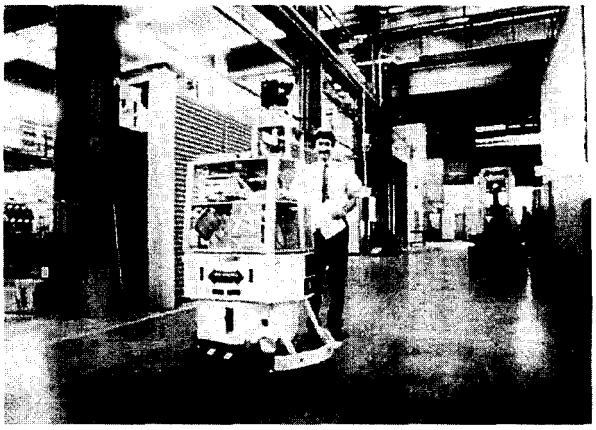


Fig. 1. The ARK industrial prototype robot within its operating environment.

The ARK robot must navigate through its environment autonomously and cannot rely on modifications to its environment such as the addition of beacons [19], 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 for pose maintenance. 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 requirements, 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 develops 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 systems could be developed.

2. Operational constraints

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 operational 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. The robot does not need to manipulate its environment, only carry mission-specific sensors. Thus the ARK robot does not have to concern itself with manipulation, only mobility.

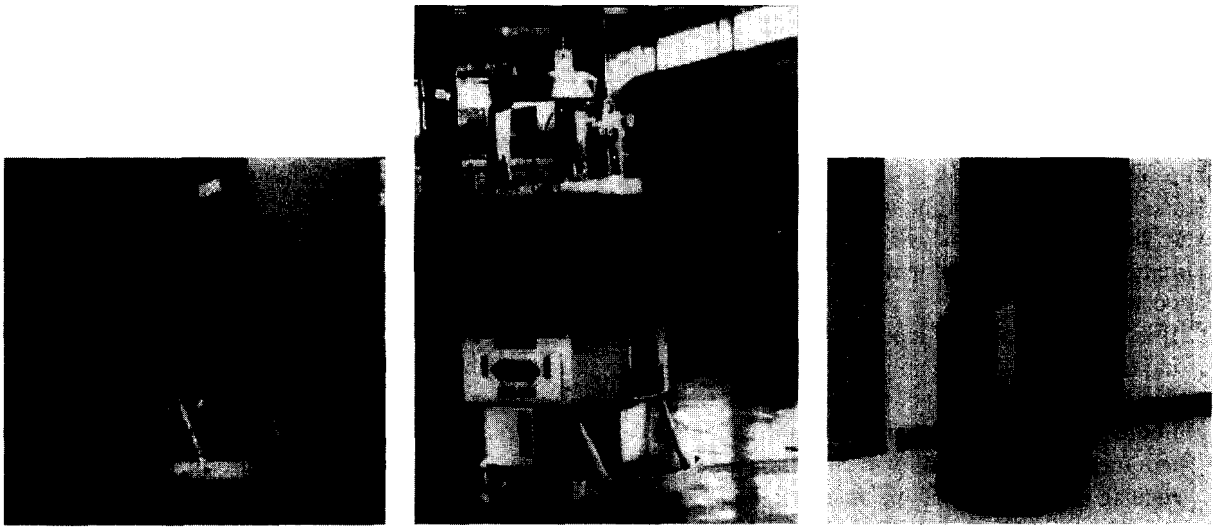
3. Overview of the ARK prototypes

It was clear from very early in the ARK project that it would be essential to build at least two versions of the robot. An initial prototype which could be used as a test platform and an industrial device to be deployed at AECL. The initial test platform would not be a fully functional device but would be constructed so that various modules could be tested independently and different solutions to technical problems could be evaluated.

In order to evaluate different approaches to the technical problems that a robot deployed in an industrial environment would face, the ARK project eventually constructed three robotic prototypes (see Fig. 2). At the University of Toronto, ARK-l was used as an initial testbed on which sensors and algorithms were tested that were ultimately included in ARK-2, the industrial prototype. For ARK-l, 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-l and ARK-2 are based on the Cybermotion Navmaster platform, while ARK-lite is based on the Nomad 200. [1]

The main hardware components of the ARK-l robot are a Navmaster mobile platform from Cybermotion, a robotic sensor head 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

[1] For details on these hardware platforms see http://www.robots.com and http://www.cybermotion.com.



(a) (b) (c)

Fig. 2. The ARK robots. The ARK-I 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.

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 [25]).

The ARK- 1 robot communicates with a network of host computers via an 8-channel remote serial link. (The link simulates eight different serial channels.) 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 the radio link to the host computer network. The computers in the network analyse these data, and generate commands for individual units of the robot (platform, head, sonar controllers, range-finder). The on-board computers perform limited 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 radio 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-l. In addition, ARK-2 is equipped with a wireless video link which runs independently of the wireless Ethernet. The wireless data 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.

The ARK robots rely on pre-generated Cartesian maps of their operational environment. The maps contain the locations of known static obstacles as well as points of interest such as the locations of visual landmarks. These maps are hand coded and form a key

component of the mobile robot system. They are used by the path planner to represent the known environmental obstacles as well as cuing the robot about the expected location of known visual landmarks. Maintaining the robot's pose with respect to the global map and dealing with objects which are not properly represented on the map are two key sensing problems which must be addressed by the robots.

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 such as Kalman filtering as well as other more ad hoc algorithms to use visual measurements to correct the robot's global position. The efficacy of the pose update/estimation process is of course a function of the accuracy of the pose measurement process. Vision alone is a poor mechanism for constraining the pose of a robot based on sightings of distant landmarks. Although the azimuth and elevation of a landmark can be used to determine distance to a landmark, this 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. *Laser Eye: A 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-l and ARK-2 robots. This sensor provides colour images and a single range measurement to distances up to 100 m which are typical for the industrial environment. ***Laser Eye*** is a combined range/video sensor consisting of a camera and a laser range-finder [8]. The laser 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 infrared 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 Fig. 3). The head can tilt in any direction between 65" below and 95" degrees above the horizon and the panning range covers 360". The head can rotate with speeds exceeding 180°/s. An initial prototype was used on ARK-l and in early experiments on ARK-2. A production version of the head was constructed at AECL and appears on ARK-2 in Fig. 3(b).

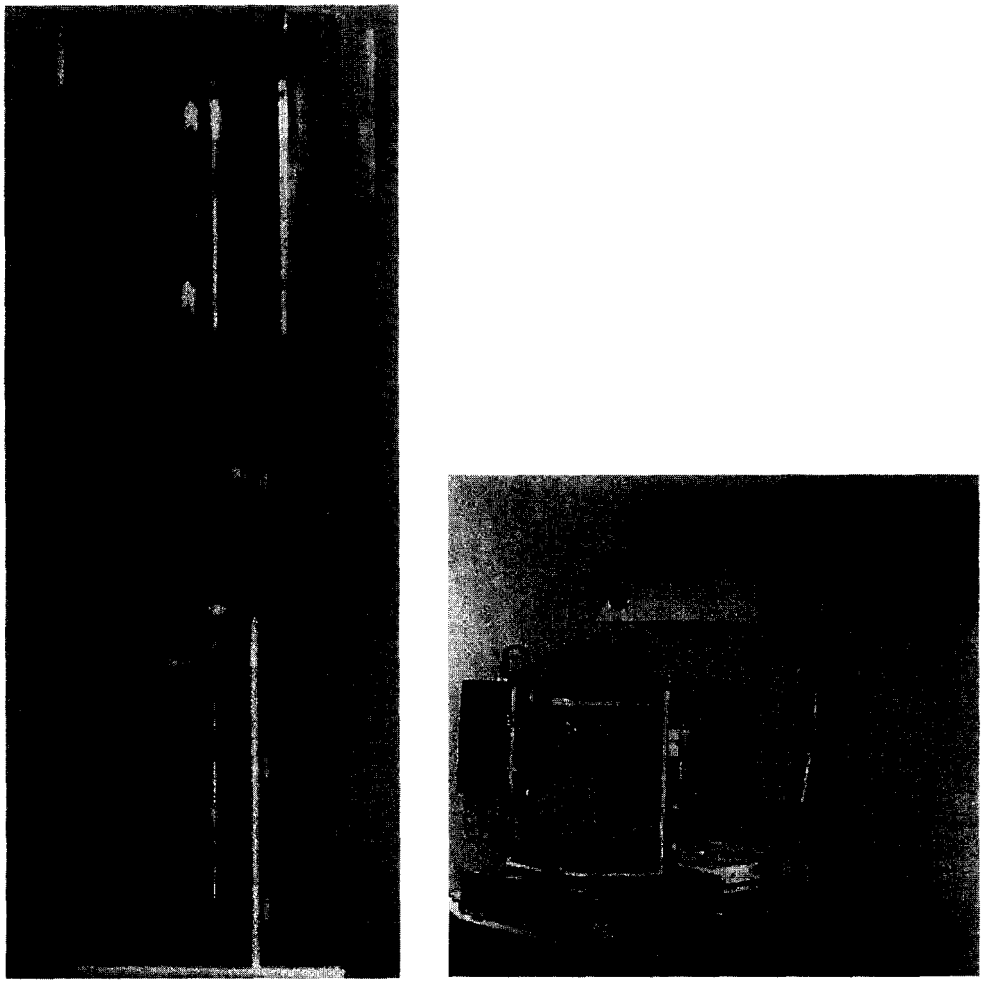
The range-finder within ***Laser Eye*** measures distance to an object in the centre of the camera field of view, In the university version of ***Laser Eye*** the camera optical axis and that of the range-finder were made coincident using a hot mirror (one that reflects infrared and transmits visible 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 infrared 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 s depending on the selected accuracy. [3] The time required to point the head in a new direction depends on the required rotation. The laser beam divergence is less than 5 m rad. This corresponds to a laser spot of three pixels in diameter for an image digitized in a 512 x 5 12 grid and for the wide setting of the zoom lens (45°). For the narrow setting of the zoom lens (4.5") 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.

[2] Model G150 made by Optech Systems Ltd., North York, Ont.

[3] More recent versions of the laser unit operate at higher speeds.



(a) (b)

Fig. 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. Fig. 2(a) shows this sensor mounted on ARK-l. (a) The first university prototype with coincident axes. (b) The industrial unit without coincident axes mounted on ARK-2. It is in the left half of the image under the radiation detector. The zoom and focus controlled lens is above the laser range finder. The video camera is hidden behind the lens. On one side of the rotating part is the control unit of the miniature video camera, and on the other the tilt motor. Data and control signals are transmitted via a slip ring at the base of the rotating part.

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 a laser scan obtained from a known position.

Fig. 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

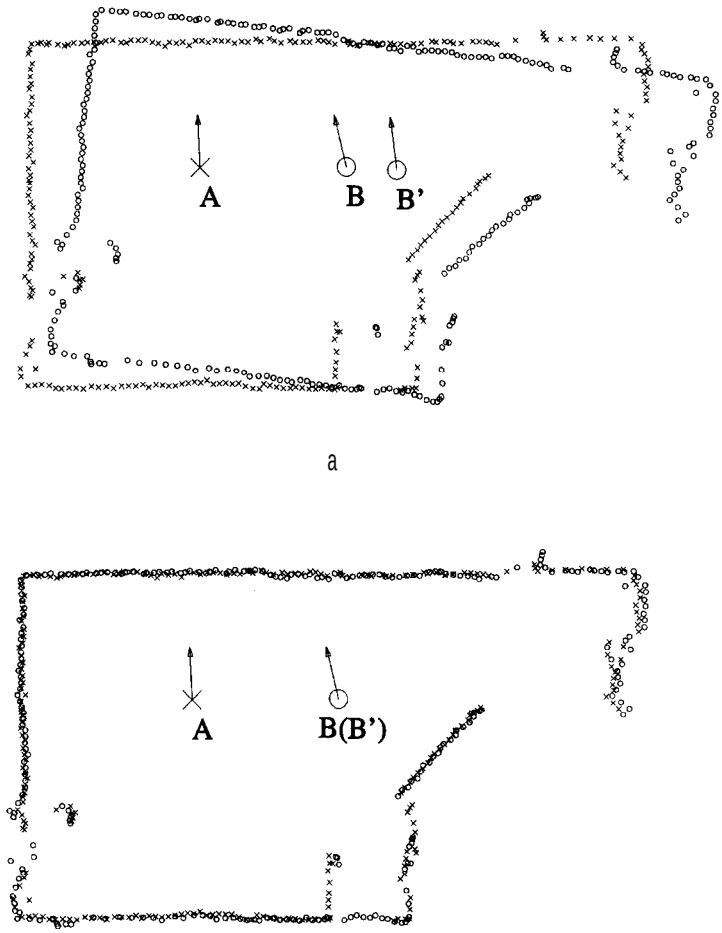


Fig. 4. Alignment of laser scans to obtain the initial robot pose. (a) The superposition of two laser scans taken within a university "office" setting with good wall structure, the one aligned with the page is taken from a known position, and a second scan from an unknown position. (b) The superposition after application of the correction algorithm.

location, a second scan from an unknown location and an initial guess of the robot's 'unknown" position, a non-linear, robust statistical technique [15] 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 s 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 [14].

This particular technique works well in regions with strong wall structure – university research labs are almost ideal – and also works in selected regions of industrial laboratories. Although this technique cannot be applied in arbitrary locations in an industrial environment, it can be applied in selected regions for initial pose estimation.

4.2.2. *Landmark recognition and tracking*

In normal operation, landmark recognition and subsequent measurement of distance, 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 to the landmark recognition problem. ARK-2 uses a generalized template matching technique applied to grey-scale images, while ARK-l 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. Visual searching for objects requires scanning the environment or checking expected locations with a camera or even moving the robot. 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 and pose of the landmark. Uncertainty in the robot's position requires the selection of a wide field of view for the camera. An attention mechanism that selects some "interesting" locations in an image or environment speeds up and simplifies the search. Features such as intensity, colour, motion and the presence of significant edges can be used to focus attention. Once candidate locations have been selected, each of them can be inspected closely to verify the presence of the target object.

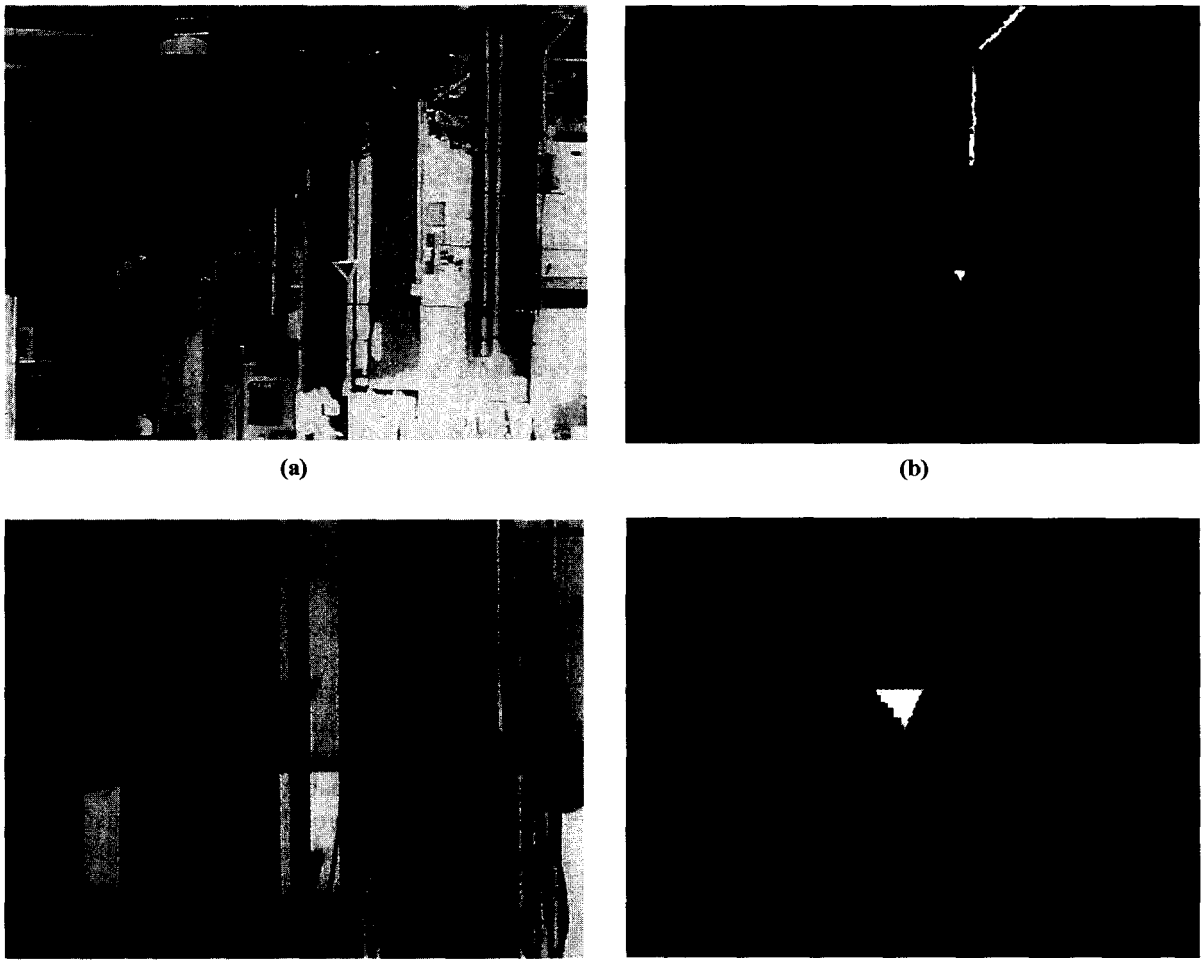
Colour can be used to identify possible candidates in an image. The ARK colour classification scheme consists of an off-line training phase and an on-line classification of pixels using a real-time image processor. Colour information is used for pixelwise classification of images and assigning pixels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

In the off-line training phase, the training sets consist of objects of interest in their natural environment under different illumination conditions. Each of the pixels in the training set is described by its hue, saturation and intensity (HSI) obtained from the measured RGB values. The training data are clustered in the HSI space using the k-means clustering algorithm. To speed up the clustering process it is performed on resampled data sets first and then these results are used to seed the clustering at higher resolutions. After clustering the user assigns individual clusters to classes that correspond to objects of interest. The assigned clusters are used by a classification algorithm to create look up tables. High resolution representation of the colour space is not necessary and it may also exceed hardware capabilities of many vision systems. The Datacube MV20 processor used in the project supports LUT of up to 64 K in size, so the input data are truncated to fit this size (5 bits for the red channel, 6 bits for the green, and 5 for the blue one).

During the on-line classification the input RGB vector is converted to a single index for each pixel. This index points to a location in the LUT with the precomputed classification result. This allows for real time operation. The resulting images contain blobs corresponding to detected objects (colour classes) and some artifacts. The artifacts are eliminated by performing morphological filtration and reconstruction. An original colour image obtained in the industrial bay at AECL is shown in Fig. 5(a) (intensity is shown here); the image contains a red triangle (a fire extinguisher sign), that the ARK robot used as a landmark. Detected red blobs are shown in Fig. 5(b): one of the blobs correspond to the triangle but there are other that correspond to red painted pipes. Determining which of the detected blobs correspond to the true sign could be made at this point using a number of different techniques. One such method would involve calculating shape properties and matching this metric with the expected values. 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 point the robotic head at each candidate in turn and then acquire and process this new set of images.

Each detected candidate is described by a set of parameters that define its position in the image, and the size and location of its bounding window. The new orientation of the head is calculated from a kinematic model of the head and the new setting for zoom is selected in such a way that the blob of interest is fully included in the new. Fig. 5(c) shows a zoomed image of one the candidates. The detected red triangle is displayed in Fig. 5(d). The actual size of the object is calculated using the range measurement to the object, its size in image coordinates and the size of the field of view.

Correlation-based landmark detection. The landmark-finding module used in ARK-2 is based on



(a) (b) (c) (d)

Fig. 5. Colour

performing a multi-resolution normalized correlation between a query image (from the robot's current position) and a ***reprqjection*** of the stored "3D grey-level surface" representing the landmark. The grey-level surface is a pyramid consisting of registered grey-level images of the landmark obtained at multiple resolutions, and the estimated depth of each pixel in 3D 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 from the robot's current pose in order to match it successfully in the query image.

Fig. 6 illustrates the geometry used for the landmark reprojection. When a landmark is first learned, a coarse range scan is performed 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. Depth values are interpolated for each pixel in the image, so that each pixel may be assigned a position in 3D space. The positions are stored in the coordinate system of the

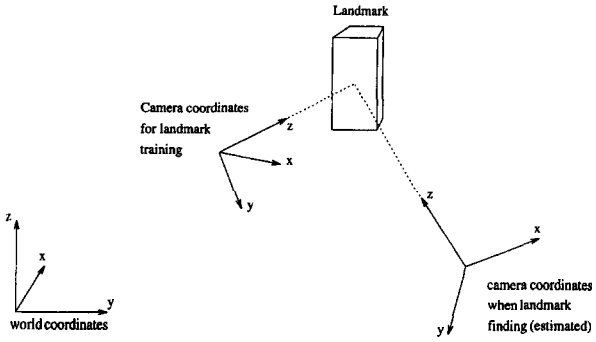


Fig. 6. Landmark reprojection geometry.

camera at the position at which the landmark is learned.

When a landmark is to be found for pose correction, the approximate pose 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. The grey-level values at selected image locations are interpolated 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 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 its safety. 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. These "standard" safety sensors were augmented with two other safety-based sensor systems; sensor algorithms to detect anomalies in the floor planes and to examine unmapped regions or regions containing unexpected obstacles.

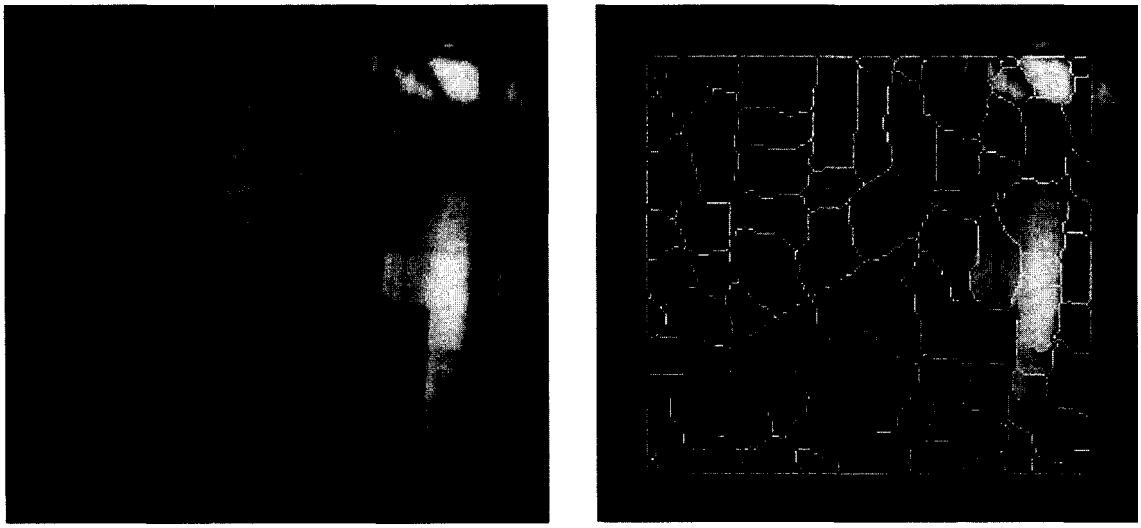
5.1. Floor anomaly detection

The floor of an industrial environment is very complex. The AECL bay, for example, contains drainage ditches (which can be open), cables, ducts, etc., which are temporally varying structures that can prevent the safe passage of the robot. The drainage ditches in the AECL bay would simply cause the robot to fall into them and tip over, resulting in serious damage to the robot, if they are not sensed and avoided. Before moving the robot onto a particular piece of the floor it is important to ensure that the floor is traversable.

Three different approaches to floor anomaly detection were considered in the ARK project: using *Laser Eye* to probe the floor in front of the robot; using passive stereo vision to localize the floor; and using a laser line striper to verify deviations from the floor plane. These approaches are described below. Although all three technologies were examined, the *Laser Eye* approach was deployed on the industrial prototype due to its lower incremental cost – it does not require additional onboard sensors.

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 traversable is 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 mechanism for 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



(a) (b)

Fig. 7. Hallway scene.

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 [23]. 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.). Fig. 7 shows an image of a hallway scene and its segmentation.

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 points 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 examined to check whether it belongs to the floor. The verification process uses the distance from the reference floor plane to accept or reject regions. Fig. 8 shows the top view of floor regions (in white) detected in the scene in Fig. 7. Fig. 8 is rotated with respect to Fig. 7. In Fig. 8 the robot is located in the lower left-hand comer. The region marked in grey represents the section of the floor which is not consistent with the flat floor model.

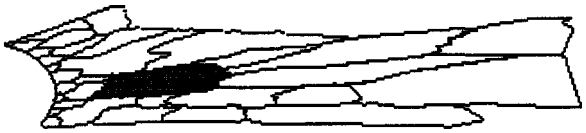


Fig. 8. Floor regions detected in Fig. 7. This image is rotated relative to Fig. 7. The semi-circular edge in the lower left-hand comer corresponds to the bottom row of Fig. 7.

Taking more than one range measurement makes it possible to reconstruct three-dimensional scenes and to verify the planarity assumption per region. The three-dimensional model shown in Fig. 9 was created by taking five range measurements per region. Positions of the ranged points in a Cartesian world coordinate system were computed by converting from the spherical coordinate system of the head (range, and azimuth and elevation of the head). Planes for individual

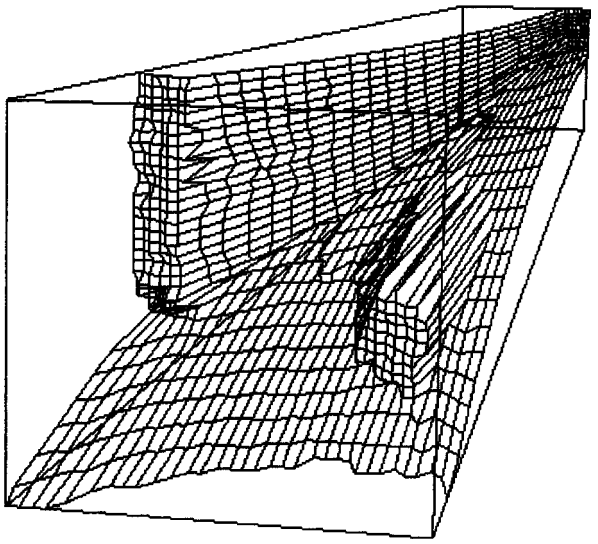


Fig. 9. Three-dimensional model for the hallway scene.

regions were estimated using a least squared error criterion.

5.1.2. *Floor anomaly detection using stereo vision*

Another approach to floor anomaly detection 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 modelled 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.

Fig. 10(a) shows a sample stereo pair of the floor cluttered with obstacles. Fig. 10(b) 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 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*

The third approach to floor anomaly detection 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. Fig. 11 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”.

Each of these three floor anomaly detection proved effective and the BIRIS based approach even operates at video rates. Nevertheless, the floor anomaly detection process using combined vision-range measurements was chosen for deployment on ARK-2 due to its ability to operate without the mounting of additional sensors onboard the robot.

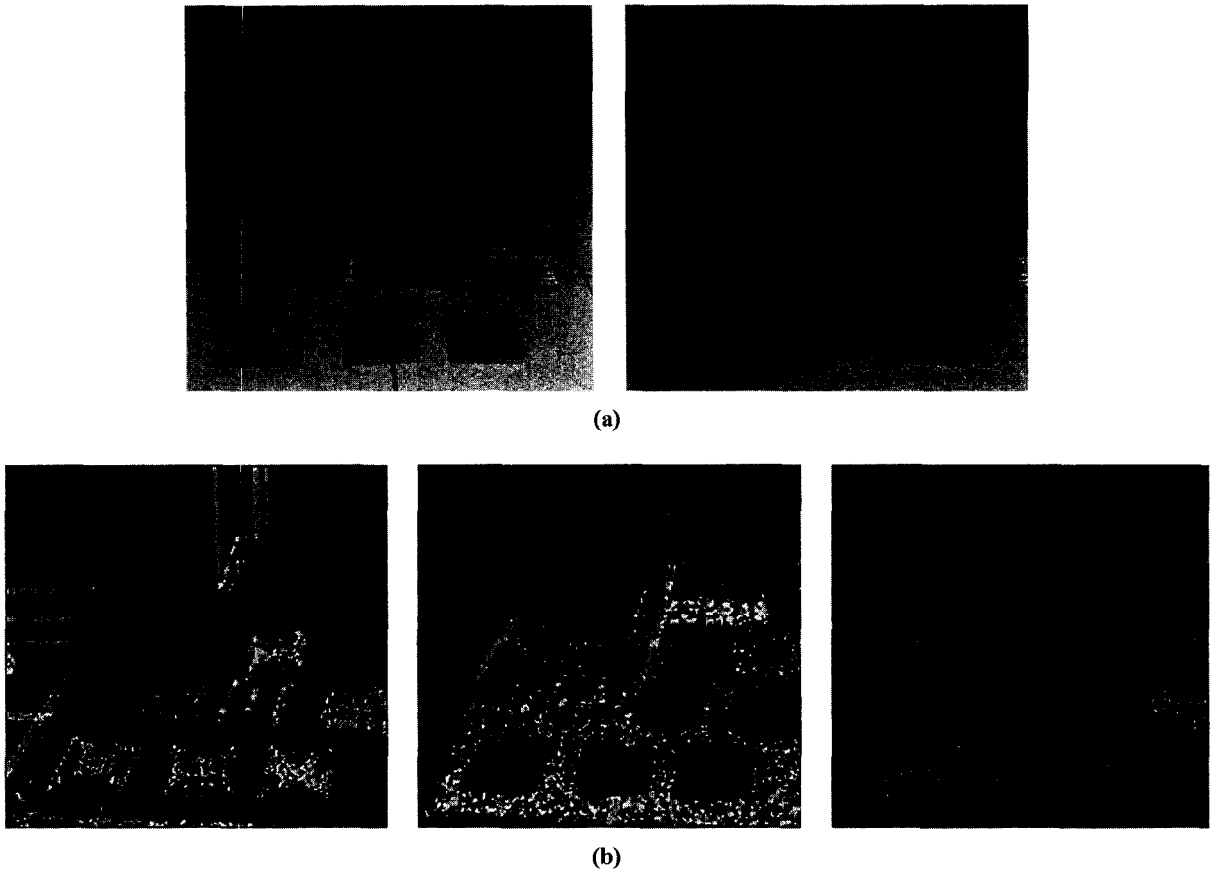


Fig. 10. Stereo-based floor anomaly detection.

5.2. *Attention-based space segmentation*

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 determine if they are traversable. This would occur in the field when exploring the environment after some disaster had occurred. Thus one additional task to which *Laser Eye* 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 traversable.

One mechanism for determining this depth map is to densely sample 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 depth measurements. Fortunately, visual image data can be used to plan where to point the range-finder [6–8].

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. Under-segmentation can cause potential problems as it requires additional depth measurements to split the region along a depth discontinuity. On the other hand, 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.

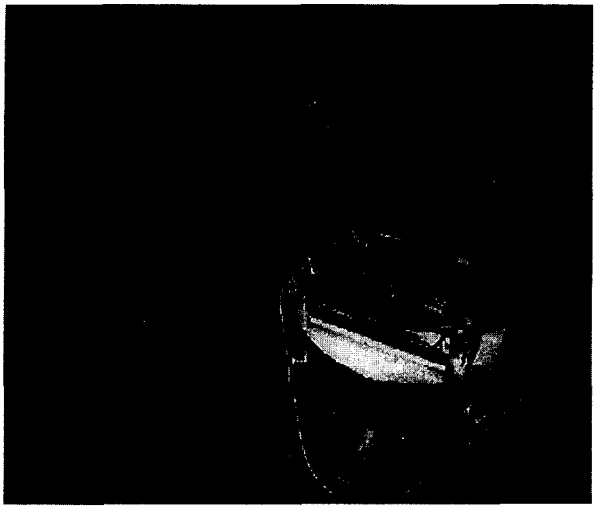


Fig. 11. BIRIS-based Floor Anomaly Detector mounted on the BIRIS a.

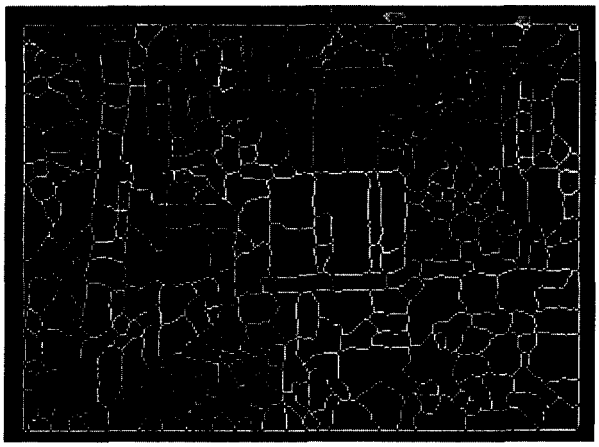


Fig. 12. Initial segmentation of the AECL.

For

tion based on image intensity only contains about 200 regions. Dense range maps can be created by taking one measurement per region and imposing arbitrary orientation of each region (e.g., orthogonal to the robot

region and assuming that the region is planar, it is possible to estimate the region

create the dense range map from 64 K samples (256 x

operating in real-time, this may still be too slow. If we look at the intensity image ourselves, it appears that a few range measurements, taken in the "right" directions, could provide the essential information for a specific task. An appropriate attention-based algorithm

location of salient feature points in the image; (ii) selection of salient features; and

specific parts of the image. This function represents

system, for example, data in the centre or below the

of the camera image. Representing the segmented im-

regions and boundaries in the graph and for access to

colour, intensity, texture descriptors, and their size and shape. The boundaries gions are described by their size, shape, orientation and contrast between regions on of winners, in the "Winner Take All" scheme [21], 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 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.

Fig. 13 shows the attended receptive fields and the path of 10 saccadic movements between regions of

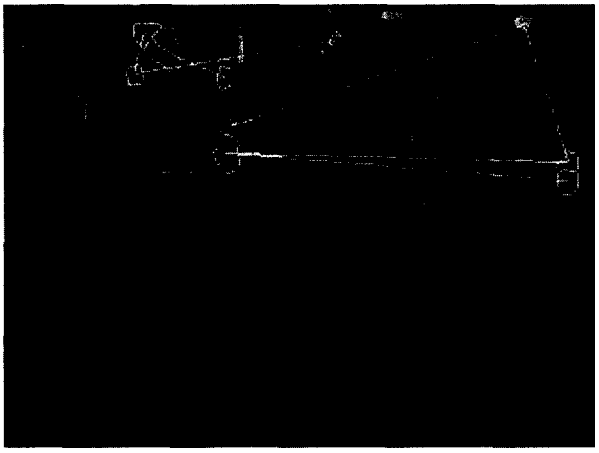


Fig. 13. Selection process for uniformly biased model.

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 software architecture consists of two levels: a discrete high level system and a reactive low level system. The high level is responsible for planning robot actions, global path planning, selecting landmarks for sighting and user interactions. 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 Fig. 14).

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, ARK takes advantage of the best of both paradigms.

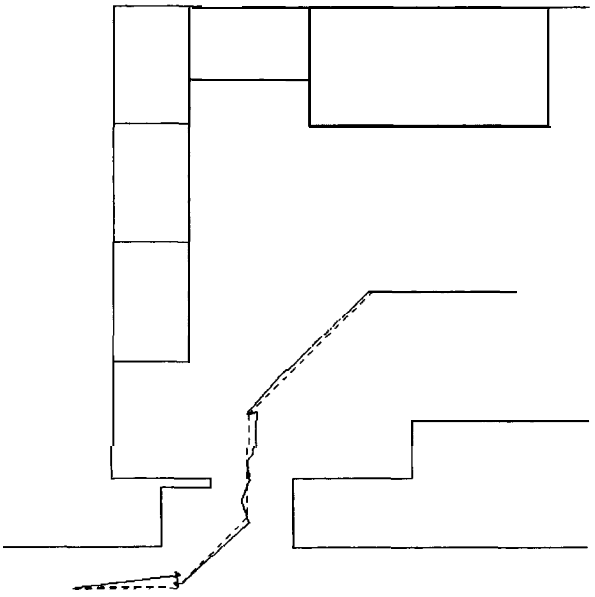


Fig. 14. Reactive robot control. The robot moves from the upper left to lower right in the figure. The commanded path is given by a dotted line while the actual path followed is drawn as a solid line. Modelled environmental obstacles are also shown.

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-l'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, ARK-lite implemented a sophisticated reactive control structure which is described below. The industrial prototype ARK-2 relies on a modified version of the ARK-l control architecture but includes a planning module to allow the robot to navigate around small unexpected obstacles.

6.1. Map representation for 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-l and ARK-lite use a configuration-space representation in which the 10 cmx 10 cm 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-l 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 in theory this may cause the only possible path between two points to be missed by the search, we have found that in practice this approach leads to substantial pruning of the space of paths 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 position of each mesh point in the path search is maintained in a data structure, to allow a 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. *Pose 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 pose fixes are done based on pre-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 Fig. 15, 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 exists 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 ***Laser Eye,*** 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 horizontal plane through the range finder then a second estimate of range is available from the elevation and a consistency check can be performed between these two estimates. Since it is possible to predict the pan angle separating the two landmarks as seen from the robot's corrected position,

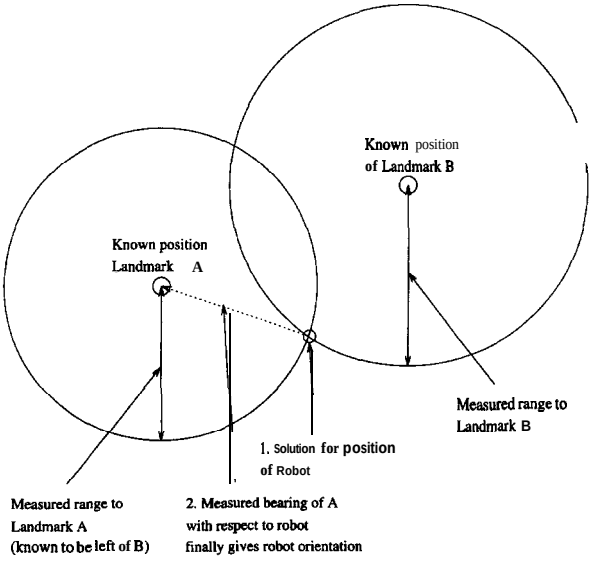
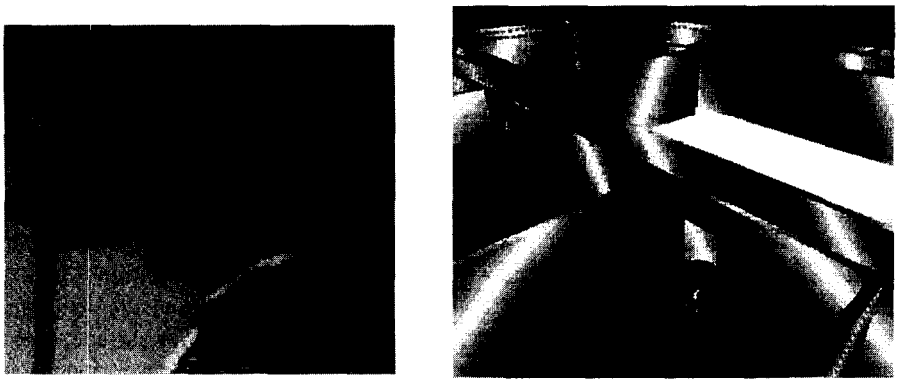


Fig. 15. ARK-2 navigation geometry.



(a) (b)

Fig. 16. Immersive user interface.

an additional consistency check on the pose correction is 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-l 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-lite relies on a more reactive low-level control mechanism [18] 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 froim 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 comer. The move behaviour steers the robot towards a precomputed goal position.

6.4. *A 3D immersive display for robotic telecontrol*

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 (see Fig. 16). 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.

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 Corp. HMD[4] which also provides stereo sound to the operator. Six degree of freedom (DOF) head tracking is accomplished via an Ascention Flock of Birds head tracker. [5] The operator is also equipped with a six DOF Cyberman three button joystick to provide additional input control. Video is generated by an SGI Indigo2 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 640 x 480 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 centre 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 centre 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" 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 centre of view. i.e., the operator simply looks straight at the object of interest. A cross-hair is always displayed in the centre of the display to aid the operator determine which interaction object is currently receiving input focus.

The immersive interface operates with the same latency as the $2\frac{1}{2}$D interface although it requires a more sophisticated graphics workstation. The $2\frac{1}{2}$D interface is more accurate in that the user can specify goal locations by providing exact (x, y, θ) goal positions, but the point and click mode of the $2\frac{1}{2}$D interface is roughly as accurate as the immersive display for specifying goal locations.

[4] For details see http://www.mbnet.mb.ca/vr/

[5] For details see http://www.ascension_tech.com

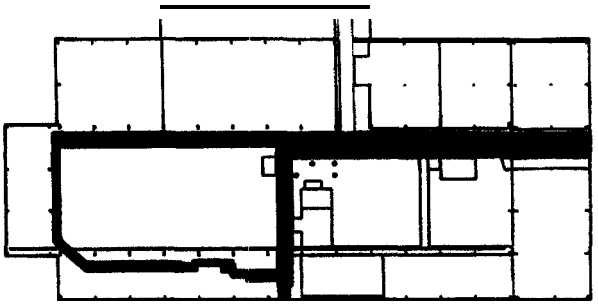


Fig. 17. Map of the AECL bay showing the ARK-2 robot's operational environment in black. The bay is approximately 4OOft long. The robot is capable of point to point motion throughout its operation region.

7. Experimental validation

The ARK-2 robot's operational environment is the AECL bay (see Figs. 1, 5 and 17). The robot was assigned the "main corridor" – the large relatively uncluttered region which stretches the entire length of the bay – as well as the "side corridor" and a complex region in the lower left-hand comer of Fig. 17 as its operating environment. Pre-existing visual features in the environment including items such as pipe joints, fire extinguisher markings, were identified and plotted on the robot's map of the environment. Given a sufficiently rich set of visual landmarks, the robot was successful in performing point to point navigation throughout this region.

Some experimentation was required in order to determine an appropriate set of landmarks. For example, in the region corresponding to the lower left-hand comer of Fig. 17, many visual landmarks had to be chosen due to the existence of dense pipe structure which obscures the view of landmarks in this region (although the structure does provide a large number of candidate visual landmarks). Visual target density was much lower in the main corridor due to the uncluttered nature of the region.

8. Discussion

In order to effectively deploy a mobile robot in an industrial environment, the robot 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/inspection tasks 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 ideal 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 unexpected 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 modem 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 concentrations. In addition to meeting these technical goals, research undertaken as part of the project has led to advances in general mobile robot systems [22], image understanding [11,16,24], system control [20], and immersive user interfaces [1].

Acknowledgements

Funding for this work was provided, in part, by the ARK (Autonomous Robot for a Known environment) Project, which received 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, 0. Sanbekkhaug, R. R. Service, D. Terzopoulos, K. Tran, and V. Wu.

References

[1] N. Aucoin, 0. Sandbekkaug, M. Jenkin, An immersive 3d user interface for mobile robot control, in: Proc. IASTED International Conference on Application of Control and Robotics, Orlando, FL, 1996, pp. 1–4.

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

[3] R.A. Brooks, A robust layered control system for a mobile robot, IEEE J. Robot and Autom. 2 (1986) 14-23.

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

[5] 0. 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, SPIE – The International Society for Optical Engineering, 1993, pp. 92-99.

[7] P. Jasiobedzki, B. Down, V. Wu, Active object detection using colour, in: C. Archibald, P. Kwok (Eds.), Research in Computer and Robot Vision, World Scientific Press, Singapore, 1995, pp. 37-52.

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

[9] M. Jenkin, E. Milios, P. Jasiobedzki, N. Bains, K. Tran, Global navigation for ARK, in: Proc. IEEE/RSJ IROS, Yokohama, Japan, 1993.

[10] M.R.M. Jenkin, A. Japson, Detecting floor anomalies, in: Proceedings of the British Machine Vision Conference, 1994, pp. 731-740.

[11] A. Jepson, M. Black, Mixture models for optical flow computation, in: I. Cox, P. Hansen, B. Julesz (Eds.), Proceedings of the DIMACS Workshop on Partitioning Data Sets: With Applications to Psychology, Vision and Target Tracking, AMS, Providence, RI, 1995, pp. 271-286.

[12] A.D. Jepson, M.J. Black, Mixture models for optical flow computation, in: Proceedings of the IEEE CVPR, New York, 1993, pp. 60-761.

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

[14] F. Lu, E. Milios, Globally consistent range scan alignment for environment maping, Autonomous Robots 4 (4) (1997) 333-349.

[15] F. Lu, E. Milios, Robert pose estimation in unknown environments by matching 2d range scans, Journal of Intelligent and Robotics Systems 18 (1997) 249-250.

[16] W.J. MacLean, A. Jepson, R. Frecker, Recovery of egomotion and segmentation of independent object motion using the em-algorithm, in: Proceedings of the British Machine Vision Conference, 1994, pp. 175-184.

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

[18] M. Robinson, M. Jenkin. Reactive control of a mobile robot, in: C. Archibald, P. Kwok (Eds.), Research in Computer and Robot Vision, World Scientific Press, Singapore, 1995, pp. 55–70.

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

[20] J.K. Tsotsos, Intelligent control for perceptually attentive agents: The S* proposal, Robotics and Autonomous Systems 21 (1997) 5-21.

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

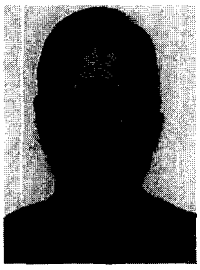
[22] J.K. Tsotsos, G. Verghese, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflo, S. Stevenson, M. Black,

D. Metaxas, S. Culhance, Y. Ye, R. Mann, PLAYBOT: A Visually-Guided Robot for Physically Disabled Children, Image and Vision Computing, to appear.

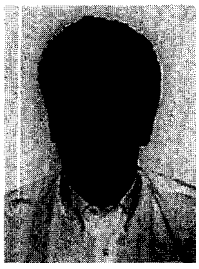
[23] L. Vincent, Watersheds in digital spaces: An efficient algorithm based on immersion simulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (6) (1991) 583-598.

[24] Z. Wang, A. Jepson, A new closed-form solution for absolute orientation, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Seattle, WA, 1994, pp. 129-134.

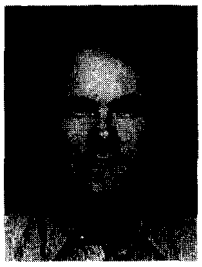
[25] D. Wilkes, G. Dudek, M. Jenkin, E. Milios, Multi-transducer sonar interpretation, in: Proc. IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993, pp. 392-397.



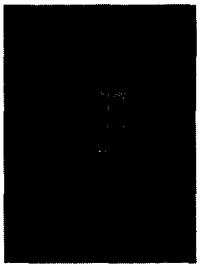
Bruce Nickerson received his Ph.D. in Physics from the University of Toronto in 1976. At present, he is a Principal Research Scientist in the Smart Systems Department at Ontario Hydro Technologies and is technically involved with programs in high voltage power system SCADA alarm message expert systems and in robotic systems for piping inspection.



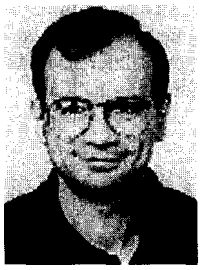
Piotr Jasiobedzki received his Ph.D. for research in Computer Vision and Robotics from Warsaw University of Technology (Poland) in 1986. Since 1995, he has been with SPAR Aerospace, Space Systems Division, where he is responsible for projects involving 3D computer vision and object recognition. Before joining SPAR, he worked as a scientist at the University of Manchester (UK) and University of Toronto (Canada). His research interests include stereo, active vision, sensor fusion, and robot control and navigation.



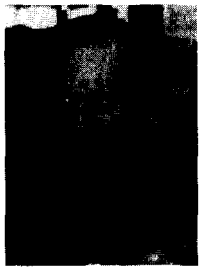
David Wilkes received his Ph.D. in Computer Science from the University of Toronto and has worked in industry on a variety of research and development projects. He is currently president of Wilkes Associates, a software and firmware contracting and consulting company specializing in product development.



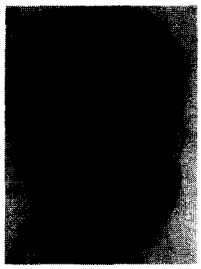
Michael Jenkin received his Ph.D. in Computer Science from the University of Toronto in 1988. He is currently the Chair of the Department of Computer Science, York University and has active research programs in Computer Vision, Mobile Robotics and Virtual Reality.



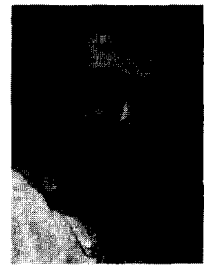
Evangelos Miios received a degree in Electrical Engineering from the National Technical University of Athens, Greece, and Master's and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA. While at MIT, he was a member of Digital Signal Processing group and he worked on acoustic signal interpretation problems at the MIT Lincoln Laboratory. He held a research faculty position at the Artificial Intelligence Group, University of Toronto (1986–1991), and was visiting professor at the Research Institute for Applied Knowledge Processing (FAW) in Ulm, Germany (1990). Since 1991, he has been an Associate Professor of Computer Science at York University. His research interests are spatial reasoning for mobile robots, shape representation, and computational auditory scene analysis.



John K. Tsotsos was born in Windsor, Ontario. He received his Ph.D. in 1980 from the University of Toronto in Computer Science. Currently he is a Professor in that department and maintains a status Associate Professorship in the University's Department of Medicine. He has served on numerous conference committees and on the editorial boards of Computer Vision and Image Understanding, Computational Intelligence and AI & Medicine. His research focuses on biologically plausible models of visual attention, the development of a visually guided robot to assist physically disabled children and perceptually guided robot control mechanisms.



Allan Jepson received a B.Sc. degree in Mathematics from the University of British Columbia in 1976. He then went to the California Institute of Technology, where in 1980 he completed his Ph.D. in Applied Mathematics. He spent two years as a postdoctoral fellow at Stanford University in the Mathematics Department, and then joined the faculty of the Department of Computer Science at the University of Toronto. From 1989 to 1995 he was named a Scholar of the Canadian Institute of Advanced Research. His current research interests include various aspects of computer vision and perceptual inference (see http: / /www.cs.toronto.edu/-jepson for more information).



Narinder Bains received his Masters degree from Warwick University in UK 1984. He was employed with Austin Rover Cars, Longbridge, UK, implementing robotics and vision systems. He also worked at SPAR Aerospace on the SPDM (Special Purpose Dexterous Manipulator) project for the Space Station Freedom. He is currently Manager, Station Maintenance at AECL and is primarily involved with remote handling and inspection systems at nuclear power plants.