Machine Vision and Applications (1997) 10:159–173

Machine Vision and Applications

Mobile robot navigation and scene modeling using stereo fish-eye lens system

Shishir Shah, J. K. Aggarwal

Computer and Vision Research Center, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712-1084, USA

Received: 25 September 1996 / Accepted: 20 October 1996

Abstract. We present an autonomous mobile robot navigation system using stereo fish-eye lenses for navigation in an indoor structured environment and for generating a model of the imaged scene. The system estimates the three-dimensional (3D) position of significant features in the scene, and by estimating its relative position to the features, navigates through narrow passages and makes turns at corridor ends. Fish-eye lenses are used to provide a large field of view, which images objects close to the robot and helps in making smooth transitions in the direction of motion. Calibration is performed for the lens-camera setup and the distortion is corrected to obtain accurate quantitative measurements. A vision-based algorithm that uses the vanishing points of extracted segments from a scene in a few 3D orientations provides an accurate estimate of the robot orientation. This is used, in addition to 3D recovery via stereo correspondence, to maintain the robot motion in a purely translational path, as well as to remove the effects of any drifts from this path from each acquired image. Horizontal segments are used as a qualitative estimate of change in the motion direction and correspondence of vertical segment provides precise 3D information about objects close to the robot. Assuming detected linear edges in the scene as boundaries of planar surfaces, the 3D model of the scene is generated. The robot system is implemented and tested in a structured environment at our research center. Results from the robot navigation in real environments are presented and discussed.

Key words: Motion stereo – Scene modeling – Fish-eye lens – Depth integration – Navigation

1 Introduction

The autonomous navigation of mobile robots has attracted a number of computer vision researchers over the years. A wide variety of approaches and algorithms have been proposed to tackle this complex problem. Perception and sensing becomes an integral part of computing for such robots that navigate in a previously unknown environment. They must be able to estimate the three-dimensional (3D) structure of the environment in order to perform useful tasks. The uses of such an autonomous agent range from providing access to hazardous industrial environments to battlefield surveillance vehicles. Various issues must be addressed in the design of an autonomous mobile robot, from basic scientific issues to state-of-the-art engineering techniques [11, 24]. The tasks a mobile robot must perform for successful autonomous navigation can be broadly classified as 1) sensing the environment, 2) building an environmental representation, 3) locating itself with respect to the environment, and 4) planning and executing efficient routes in the environment. Such complex tasks cannot be completely programmed a priori, thus sensing becomes critical in monitoring both the environment and the robot's own internal reckoning system.

The environment in which the robot performs various tasks determines the techniques used by the robot to navigate. Navigation in indoor structured environments has been studied by several researchers [9, 16, 40]. Different sensor modalities such as visual sensors (monocular, binocular stereo and trinocular stereo), infrared sensors, ultrasonic sensors, and laser range finders have been used to aid in the task of environment representation. The robot can also be assisted in its navigational tasks by providing it with a priori information about the environment. In the absence of a priori model of the environment, it becomes necessary to rely solely on sensor information. In navigation tasks, extraction of accurate depth information is important. Sensing has mainly been accomplished through the use of stereo or monocular image sequences. Stereo systems establish correspondence between images acquired from two well-separated cameras, while monocular systems account for small image displacements.

In this paper, we present a system for autonomous navigation based on binocular stereo using a pair of fish-eye lenses. No a priori model is assumed, but the problem is restricted to an indoor structured environment. Navigating in indoor environments can be very challenging when the environment is narrow and has corners which require sharp manuevers of the robot. Conventional visual sensors such as normal or wide-angle CCD cameras do not provide enough

Correspondence to: J. K. Aggarwal

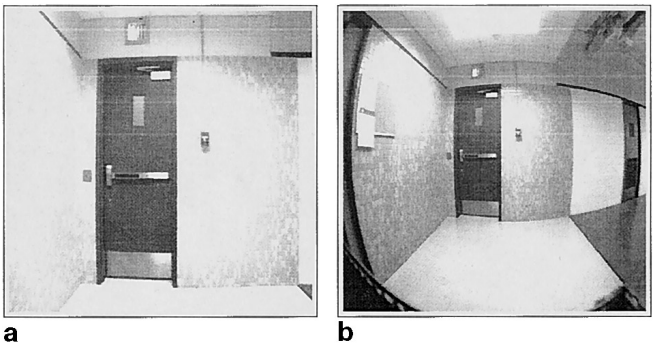


Fig. 1. a Image taken by a wide-angle lens at corridor end. **b** Fish-eye lens image taken from the same position contains more information about the corridor

information to make precise measurements close to the robot and fail to sense areas which exhibit sharp turns in the motion direction. This difference in information is shown by the two images in Fig. 1. It shows that fish-eye lenses provide a large field of view [45] and can sense valuable information close to the robot and at sharp transitions in the corridor or passage to be navigated. The system is implemented for a robot (*RoboTex*) that navigates through narrow passages and makes transitions in motion direction at sharp corners in a structured environment. The lenses are calibrated and distortion corrected before further processing. A specialized line detector is used to extract line segments in three significant 3D orientations. An accurate estimate of the robot's egomotion is obtained from the odometry, corrected by a vision-based algorithm which uses vanishing points of the detected lines to accurately estimate the robot's heading, roll, and pitch. Correspondence between the extracted features is established based on an iterative hypothesis estimation and verification procedure [47]. The estimated 3D segments are repeatedly updated under ordered constraints with detected segments from following image sequences obtained after the motion of the robot.

The rest of the paper is organized as follows: Sect. 2 provides a survey of relevant work in indoor mobile robot navigation and scene modeling. Section 3 describes the sensor system used in our work. The stereo setup and characteristics of the fish-eye lens and its advantage over conventional lenses are discussed. The process of calibration and distortion correction are briefly discussed. Section 4 discusses the navigation environment. The segmentation approach is described and the representation of the environment using 3D line segments is discussed. Section 5 describes the process of 3D sensing, modeling and navigation based on stereo. The stereo correspondence procedure is also discussed. In Sect. 6, the experimental robot, *RoboTex*, is introduced and implementation results of the proposed system are presented. Finally, conclusions are presented in Sect. 7.

2 Literature survey

Numerous techniques have been studied for mobile robot navigation in various environments. One of the earliest attempts in indoor mobile robotics was the work of Moravec [40, 41]. Moravec used a mobile platform as a test bed for experiments in visual perception and control. The robot was remotely controlled and equipped with a single video camera. The system used a slider stereovision system to obtain 3D spatial information for navigation. To obtain a stereo pair through the slider system, the camera was moved along a track after acquiring one image to generate the disparity field, which then could be transformed into depth information. The mobile robot, *Hilare*, designed by Giralt et al. [19] and Chatila and Laumond [9] used dynamic world modeling for navigation and world representation. The robot was equipped with several sensors including a video camera, a rotating sonar sensor, and a laser range finder. The robot built a world representation by incorporating any previously unknown object into a world model, as it was perceived during navigation. Hierarchical approaches using global and local model updates based on the sensor data feedback have also been suggested by Crowley [12] and Parodi [44]. Heuristic-based approaches to navigation have also been explored by Chattergy [10]. In general, most algorithms rely on 3D measurements for navigation and scene modeling, but it has also been shown that similar results can be obtained by using a single camera with assumptions regarding the 3D environment [43, 51]. Use of stereovision with line segments to recover passive 3D measurements has also been successfully used by [23, 26]. The use of trinocular stereo has also been explored by [1, 14, 17]. Different approaches that have been studied and experimented for mobile robot navigation can be classified in three broad categories: model-based approaches, landmark- or reference-based methods, and trajectory integration methods. For a detailed survey, refer to [48].

In the work by Kak et al. [28], they present the system FINALE (Fast Indoor Navigation Allowing for Location Errors). In this approach, they use a Kalman Filter for uncertainty reduction, position estimation and updating. A geometric model of the environments is assumed, and matches between landmarks from the model and the monocular images must be determined. The system determines the approximate location of landmarks in the scene by placing bounds on where one should look for landmarks in the camera image. Given the uncertainty at any location of the robot, an expected scene model can be constructed and the uncertainties represented by the Mahanabolis distance. Each edge uncertainty regions are derived based on the vertex uncertainty regions, and the search for scene correspondents in the the model can be limited in the images and Hough space. By analyzing and processing the images only within the uncertainty regions associated with the landmarks, correspondence is quickly established. With the correspondence established and using Kalman filtering, it is possible to compute the orientation and location of the robot in the environment. The authors present real-scene results and also test the accuracy of their system. Kak et al. [25] have also considered the navigation of a mobile robot in a structured scene using a CAD description and a visual camera. The general idea is to match the observed features in the environment to the CAD model and thus estimate the robot position. More recently, the work presented by Bouguet and Perona [7] describes a visual navigation system using a single camera which utilizes a special detector to detect the corners of landmarks placed in the scene. By tracking these detected features, they estimate the image flow and compute the motion parameters. They

use a recursive motion estimator and the 3D scene structure to reconstruct the actual structure of the scene. Dalmia and Trivedi [13] present a motion and stereo integration approach to recover the depth in a scene. The authors use the ability of stereo processing to acquire precise depth measurements along with the efficiency of spatial and temporal gradient (STG) analysis. STG analysis has been shown to provide depth with high processing speeds, but limited accuracy. Methods such as normalizing, cross-correlation, etc. are used for estimating the disparity value. This value is later used by the STG process to improve the efficiency of the matching process. Experiments performed validate the integration approach to estimate depth with a mean error of 3%. Lebègue and Aggarwal [31–33] have shown a monocular vision system for navigation and CAD model generation of the indoor environment. They consider line segments and consider the modeling as surface patches bounded by line edges. They assume that any indoor scene and the objects within it can be represented by linear segments oriented in a limited set of directions. They segment line segments in three 3D orientations and, based on robot motion and correspondence over subsequent frames, try to recover the robot position and the positions of 3D line segments in world space. The lines are integrated over time and uncertainty is calculated using the Kalman filtering technique. The robot updates the world representation at each step, and the final CAD model is generated when the robot has finished navigating the environment. One major drawback in this system is that, due to the limited view angle of a single wide-angle lens, the robot is not able to navigate in narrow passages or to make sharp transitions in its motion direction at hallway corners. It also cannot accurately map hollow areas in the scene. The CAD model generated by this system for a hallway is shown in Fig. 2. The major drawbacks of such a system are that it is necessary to have a wide area of space available for navigation. Further, sharp corners and turns cannot be represented with good accuracy, and the robot cannot navigate around them due to lack of information. Such limitations have provided the motivation for the work described in this paper.

3 Visual sensors

The mobile robot is equipped with two fish-eye lens cameras configured in a parallel stereo geometry. The sensors are placed so as to maintain a maximum overlap of viewed scene to establish correspondence. The lenses have a calculated focal length of 3.8 mm and the stereo pair has a baseline distance of 398 mm.

3.1 Stereo setup and image information

Using the parallel axis geometry, a disparity value, d, is determined for each matched feature as the difference in their positions on the same horizontal scan line. Any detected two-dimensional (2D) feature in an image is considered the perspective projection of a 3D feature in the scene. In general, to recover depth, two images acquired from different perspectives are used to establish the transformation relationship between the scene and its projection in the left and right images. As shown in Fig. 3, a point P, defined by coordinates (x, y, z) in 3D will project in 2D with coordinates (x_l, y_l) and (x_r, y_r) for left and right images, respectively. Knowing the baseline distance, D, which separates the two cameras, and the focal length f, we can define the perspective projections by simple algebra. Considering O to be the origin coinciding with the image center in the left camera:

$$x = x_l.D/d \ , \tag{1}$$

$$y = y_l.D/d \ , \tag{2}$$

$$z = f.D/d \ . \tag{3}$$

These equations provide the basis for deriving 3D depth from stereo images.

Fish-eye lenses provide a field of view which approximates 180° in the diagonal direction. Objects very close to the lens can be imaged with good accuracy. At the same time the large horizontal view also provides valuable information for rotational motions while navigating. This would not be possible using a normal lens. This is shown in Fig. 1. It contrasts a scene as imaged by the fish-eye lens (right), and as imaged by a wide-angle lens (left) from the same position.

3.2 Calibration and distortion correction

As seen in Fig. 1, the fish-eye lens image exhibits significant distortion of the information. To make precise quantitative measurements, it is important that the lenses be accurately calibrated and that both the extrinsic and intrinsic parameters be measured. The intrinsic parameters to be calibrated include the optical center, focal length, and one-pixel width. These are also crucial in successfully removing distortion in the fish-eye image to recover linear features. The procedure for calibration is described in [8, 45, 50]. The distortion correction is achieved by performing a nonlinear mapping of points in the image plane. A simultaneous correction of the radial and tangential offset components is performed using fifth-order polynomial. The general equations are given as:

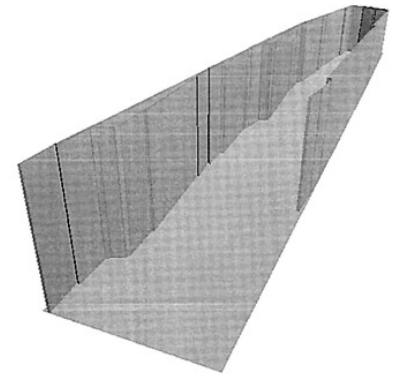
$$\theta' = a\theta + b\theta^2 + c\theta^3 + d\theta^4 + e\theta^5 \ , \tag{4}$$

$$\rho' = f\rho + g\rho^2 + h\rho^3 + i\rho^4 + j\rho^5 \ , \tag{5}$$

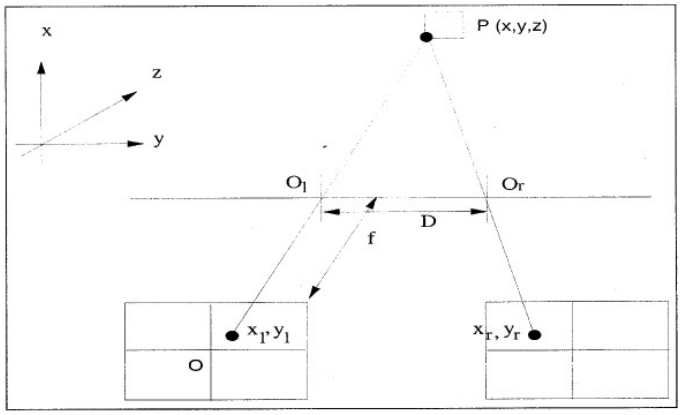
where θ' is the corrected angle, θ is the original angle, ρ' is the corrected radius, ρ is the original radius, and a through j are the distortion coefficients. The detailed algorithm is described in [45, 50]. Figure 4 displays the fish-eye image on the left, and the corrected image on the right. The black areas observed at the top and bottom of the corrected image are due to a field of view larger in the diagonal direction than the vertical direction.

4 Robot navigation

The robot navigates in indoor structured environments such as corridors, hallways, etc., based on the 3D information derived by stereo fish-eye cameras, along with the robot heading values computed using a vision-based algorithm. The procedure is described in more detail below.



2



3

4a

4b

Fig. 2. CAD model of hallway by visual navigation using a single wide angle lens

Fig. 3. Stereo system

Fig. 4. a Inherent distortion in fish-eye image. **b** The undistorted image

4.1 Environment

Visual navigation in indoor structured environments results in the representation of many interesting objects and features by planar patches bounded by linear edges. The 3D orientation of thoses edges often falls in a discrete set of possible orientations. It is seen that an environment such as a corridor is mainly composed of linear edges with particular orientations in 3D. There are three preferred orientations for linear segments that qualify as significant features. The linear edges are boundaries of opaque planar patches, such as the floor, ceiling, walls, etc. This not only presents an accurate representation, but also simplifies the computation. The particular orientations in 3D considered in our approach are the vertical and two horizontal orientations perpendicular to each other.

Under the perspective projection geometry, each 3D orientation corresponds to one vanishing point in the image plane. This is the point (possibly at infinity) where all the lines, parallel in one direction, seem to originate from. This is illustrated in Fig. 5. To segment the images and extract the line segments, a specialized line detector which uses the a priori knowledge of the locations of the vanishing points is used [29]. This process can assign the closest 3D orientation to each detected segment. In particular, with a pinhole perspective projection model, lines that are parallel in 3D will converge to a vanishing point in the 2D projection. If the orientation of the camera with respect to the scene is approximately known, one can compute the vanishing points associated with each given 3D orientation before processing the image. Line segments detected in three orientations are grouped according to their most likely 3D orientation and are considered for the correspondence problem.

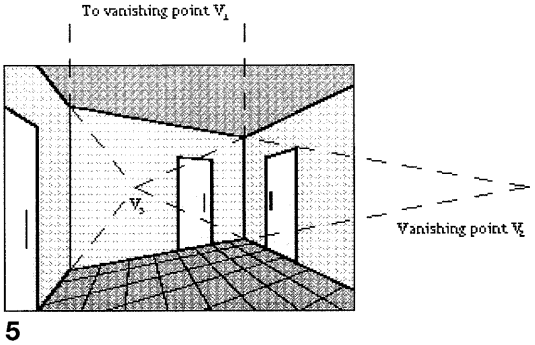
4.2 Coordinate system

The fish-eye lens cameras are mounted on the robot, thus the motion of the robot has to be defined with respect to the robot platform, and not just the camera system. In order to use quantitative information, the relationship between the robot and camera coordinate system must be known. Figure 6 shows the world and the robot coordinate systems. The cameras are mounted at two positions on the robot a fixed distance apart. It is assumed that the camera is rigidly attached to the robot. The z-axis is the optical axis of the camera. **W** represents the world coordinate system with a vertical z-axis, **R** the robot coordinate system, **C** the camera coordinate system, and **P** the image coordinate system. If the heading, roll, and pitch of the robot are given by h, r, and p, then the homogenous transformation from the world to the camera coordinate system is given by

$$H_{WC} = H_{WR} H_{RC} \ , \tag{6}$$

where $\mathbf{H_{WR}}$ is given by

$$H_{WR} = T_r . \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos p & \sin p & 0 \\ 0 & -\sin p & \cos p & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos h & \sin h & 0 & 0 \\ -\sin h & \cos h & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$



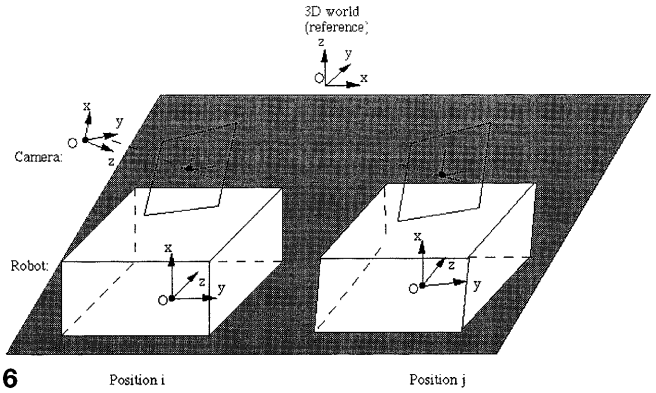


Fig. 5. Vanishing point for semantically significant 3D orientations in a typical indoor scene

Fig. 6. Coordinate systems

$$\begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

where T_r represents the transformation matrix associated with the roll of the camera mount, and x, y and z represent the relative position of the camera mount with respect to the world. Then, the perspective projection from the camera to the image plane is given by

$$\begin{bmatrix} su \\ sv \\ s \\ 1 \end{bmatrix}_P = \begin{bmatrix} \alpha_u f & u_0 & 0 & 0 \\ 0 & v_0 & -\alpha_v f & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_C, \tag{8}$$

where u and v represent the coordinates of a point on the image plane (pixels), α_u and α_v are conversion factors (pixels per unit length), u_0 and v_0 are the coordinates of the optical center of the camera (pixels), and f is the focal length (unit length). For the relationship between the robot and the camera coordinate system, given the rotation matrix **R** and the translation vector **T**, the homogenous transformation matrix, **H**, is given by

$$H = \begin{bmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

To determine the homogenous transformation matrix from the robot to the camera coordinate system, $\mathbf{H_{CR}}$, we need to know the transformations between the two camera systems, $\mathbf{H_{CP}}$, and the transformation between the two robot coordinate systems, $\mathbf{H_{RC}}$. Then, $\mathbf{H_{CR}}$ can be defined by the relation

$$H_{CR}H_{RC} = H_{CP}H_{CR} \tag{10}$$

For the above equation, $\mathbf{H_{RC}}$ is known from the robot odometry, and $\mathbf{H_{CP}}$, the camera motion, is to be computed from the two images. Having computed the intrinsic and extrinsic parameters at each location of the camera on the robot, the individual transformation matrices $\mathbf{H_{Ci}}$ and $\mathbf{H_{Cj}}$ are known. Then the transformation between the two camera systems is given by

$$H_{CP} = H_{Ci}^{-1} H_{Cj}\ . \tag{11}$$

The rotation and translation components are then simply given by

$$R_{CR}R_{RP} = R_{CP}R_{CR}\ , \tag{12}$$

$$R_{CR}T_{RP} - T_{CP} = (R_C P - I)T_{CR}\ , \tag{13}$$

where **I** is the identity matrix. The camera motion, $\mathbf{R_{CP}}$ and $\mathbf{T_{CP}}$ are given by

$$R_{CP} = R_{CR}R_{RP}R_{CR}^{-1} \tag{14}$$

and

$$T_{CP} = (I - R_{CP})T_{CR} + R_{CR}T_{RP} \tag{15}$$

4.3 Significant line detection

Significant segments in indoor scenes should be those which consider the geometric features representing the environment. For navigational purposes, line segments oriented in three particular directions prove to be useful. Line segments can be extracted by estimating the location of radical change in intensity values. Edge locations in any image can be found by considering the image to be a 2D surface and taking the second-order derivative in the x and y directions. In our implementation, we use precomputed vanishing points to link detected edgels to form a line in one of the three orientations. The vanishing points are determined from the heading information obtained by calibrating the camera parameters, and are updated based on the horizontal lines in the image. Given the three 3D orientations, all detected lines must pass through the associated vanishing point when projected. Computing the angle between the intensity gradient and the expected direction of the line in 2D, the expected line can be given by the current pixel and the vanishing point associated with the possible 3D orientation. Knowing the homogenous transformation matrices for changing the world coordinate system into the projective coordinate system, we can consider

$$\begin{bmatrix} p_{\mathrm{x}} \\ p_{\mathrm{y}} \\ p_{\mathrm{z}} \\ 0 \end{bmatrix}_{\mathrm{W}}$$

to be a nonnull vector in the 3D direction under consideration, and if

$$\begin{bmatrix} su \\ sv \\ s \\ 1 \end{bmatrix}_{\mathrm{P}} = T_{\mathrm{WP}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{\mathrm{W}} \tag{16}$$

defines the relation between a 2D point $\begin{bmatrix} u & v \end{bmatrix}^{\mathrm{T}}$ and its corresponding 3D location by the perspective projection, then

$$\begin{bmatrix} s'u' \\ s'v' \\ s' \\ 1 \end{bmatrix}_{\mathrm{P}} = T_{\mathrm{WP}} \left(\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{\mathrm{W}} + \begin{bmatrix} p_{\mathrm{x}} \\ p_{\mathrm{y}} \\ p_{\mathrm{z}} \\ 0 \end{bmatrix}_{\mathrm{W}} \right) \tag{17}$$

defines another point of the estimated 2D line. A 2D vector j in the image plane pointing to the vanishing point from the current point is then collinear to

$$\begin{bmatrix} u' - u \\ v' - v \end{bmatrix} .$$

Algebraic manipulations lead to

$$j = \begin{bmatrix} j_{\mathrm{u}} \\ j_{\mathrm{v}} \end{bmatrix} = \begin{bmatrix} a_{\mathrm{x}} - a_{\mathrm{z}}u \\ a_{\mathrm{y}} - a_{\mathrm{z}}v \end{bmatrix} , \tag{18}$$

where

$$\begin{bmatrix} a_{\mathrm{x}} \\ a_{\mathrm{y}} \\ a_{\mathrm{z}} \\ 0 \end{bmatrix} = T_{\mathrm{WP}} \begin{bmatrix} p_{\mathrm{x}} \\ p_{\mathrm{y}} \\ p_{\mathrm{z}} \\ 0 \end{bmatrix}_{\mathrm{W}} . \tag{19}$$

Note that a_{x}, a_{y}, and a_{z} need to be computed only once for each 3D orientation. A more detailed formulation and description of the algorithm can be found in [30] and [29]. The fish-eye, undistorted, and segmented images are shown in Fig. 7.

4.4 Robot pose

Robot orientation is accurately determined by considering both the odometry and the 3D position estimated by stereo correspondence. The odometers are placed on the left and right wheels of the robot and the average of their reading is taken. Due to slippage, the odometers drift without bounds over long distances and become unreliable measures. This drift is periodically corrected by a vision-based algorithm that computes the heading from the vanishing points of the extracted lines. To estimate the rotation around the world coordinate system, the vanishing points in the horizontal direction are considered. The point closest to the image center is used, and to recover the pitch, p, and heading, h, the vanishing point (u_{vp}, v_{vp}) is calculated as

$$\begin{aligned} u_{vp} &= \lim_{s\to\infty} \frac{su}{s} , \\ v_{vp} &= \lim_{s\to\infty} \frac{sv}{s} , \end{aligned} \tag{20}$$

where s, u and v are given through calibration. Now the pitch and heading can be calculated by

$$\begin{aligned} p &= \tan^{-1} \\ &\left[\frac{(H_{1,1}H_{2,0} - H_{2,2}T_{1,0}) * u_{vp} + (H_{2,1}H_{0,0} - H_{0,1}H_{2,0}) * v_{vp} + (H_{0,1}H_{1,0} - H_{1,1}T0,0)}{(H_{1,2}H_{2,0} - H_{2,2}T_{1,0}) * u_{vp} + (H_{2,2}H_{0,0} - H_{0,2}H_{2,0}) * v_{vp} + (H_{0,2}H_{1,0} - H_{1,2}H_{0,0})}\right], \\ h &= \tan^{-1}\left[\frac{H_{0,1}\cos p - H_{0,2}\sin p - (H_{2,1}\cos p - H_{2,2}\sin p) * u_{vp}}{H_{2,0}u_{vp} - H_{0,0}}\right] , \end{aligned} \tag{21}$$

where, $\mathbf{H} = \mathbf{H_{RP}}$, the transformation matrix from the robot to the image coordinate system, and $\mathbf{H_{i,j}}$ are the i, j elements of the matrix.

Although the path of the robot is calculated at each step using both the 3D depth estimate and robot pose from vanishing point, the initial estimate of motion direction is based on the estimated depth. When the depth estimated falls below a threshold, the robot must make a change in its navigating direction. This change in direction is based on the horizontal depth estimate. After making an initial rotation, the robot pose is once again estimated and corrected to move towards the vanishing point.

5 3D sensing

The mobile robot navigates based on extracted 2D features from the image scene. The parallel-axis stereo system uses a pair of fish-eye lenses and features in the two images are matched using an iterative hypothesis verification algorithm [47]. From the line correspondences and the predetermined camera parameters, an inverse perspective geometry is used to recover the 3D information. The robot then integrates the information with its pose information and calculates the motion step. The robot then moves to the next position and repeats the procedure. Already detected 3D segments are updated based on the 2D segments and their estimated 3D locations computed from the new image obtained after robot motion. The stereo matching algorithm and 3D information recovery procedure is discussed below.

5.1 Stereo matching

The analysis of stereo images is a well-established passive method for extracting the 3D structure of an imaged scene. The main objective is to recover the 3D position of detected features from their projections in 2D images. The basic principle involved in depth recovery using passive imaging is triangulation. Here, we present a stereo matching algorithm for depth recovery in indoor scenes, using a pair of fish-eye lens cameras.

5.1.1 A brief review of stereo-matching techniques

The computational stereo paradigm consists of three major steps: stereo camera modeling, feature detection, and matching. The matching process begins with the identification of the features in both images that correspond to the same point in the 3D space. This is a difficult problem, as the features extracted from the two images may be dissimilar, since they are, in fact, two perspective views taken from two different view points. Thus, further constraints have to be imposed and the matching criterion relaxed to resolve ambiguities. Generally, two broad classes of techniques have been used: feature-based techniques and area-based techniques. Feature-based solutions employ simple, geometric primitives such as line segments and planar patches [20]. Such models are appropriate for simple, well-structured environments consisting of man-made objects. These techniques generally have been more successful overall, as the matching process is much faster than the area-based techniques and there are fewer feature points to be considered [37]. The area-based

Fig. 7. a Fish-eye image. **b** The corrected image. **c** 2D lines extracted in three semantically significant 3D orientations. The dot at the center is the location of the vanishing point of the horizontal lines going into the image plane

algorithms represent depth at each pixel in the image. These techniques promise more generality; however, much remains to be done on both the mathematical and system aspects of this approach [5,36]. So far, many techniques have proved to be unsatisfactory, as they produce poorly defined matches, and thus it becomes difficult to determine when a match has been established. Area-based techniques are also highly sensitive to distortion in gray level and geometry, and are computationally very expensive.

Simple geometric features such as line segments have been commonly used as matching primitives for feature-based techniques. Edge segments alleviate the effects of positional error due to isolated points and support the constraint of edge connectivity. Medioni and Nevatia [37] use a disparity continuity constraint for their segment-based matching algorithm. Linear edge segments are extracted [42] and described by the coordinates of their endpoints, their orientation, and the average contrast in intensity along the normal of their orientation. Segments in the left and right images are matched by evaluating a merit function iteratively which minimizes the disparity difference among matched features. This algorithm implements the surface continuity constraint proposed by Marr and Poggio [35]. Another scheme proposed by Mohan, Medioni, and Nevatia [38] detects and corrects local segment matching errors based on disparity variation across linear segments. Ayache and Faverjon [3] use segments described by their midpoint, length, and orientation for stereo matching. A neighborhood graph is used to store the information regarding adjacency of segments in each image. A disparity gradient limit criterion is used to guide the global correspondence search which propagates matches within a neighborhood to recover 3D segments lying on a smooth surface patch. Their approach once again favors matches which make the 3D scene maximally smooth by maintaining the surface continuity constraint [35].

Many of the area-based techniques make use of the correlation measure to establish matches within a neighborhood of points in the given images. The Moravec interest operator [39] was used by Moravec to determine area-based correlation with a coarse-to-fine strategy to establish correspondence between points. The operator measures directional variance of image intensity in four directions around each pixel and the corresponding image is searched at various resolutions, starting from the coarsest. The position yielding a high correlation is enlarged to the next finer resolution. Matches above a certain threshold could be accepted. Using the statistics of noise in image intensities, Gennery [18] used a high-resolution correlator to produce improved estimates of match points. It also provided an estimate of the accuracy of a match in the form of variance and covariance of the pixel coordinates of the match in the corresponding image. Another correlation-based method was used by Hannah [21], which used a modified Moravec operator to determine control points. Autocorrelation was used as a measure to evaluate an established match. In a later implementation [22], Hannah implemented a hierarchical correlation method, where images were smoothed by a Gaussian window to obtain a lower resolution image. Control points were once again picked by the modified Moravec operator and the search for a match was conducted, which would result in a maximum in normalized cross-correlation with the original point. The search was propagated up to the finest resolution, at which point the search was repeated with reversed left and right images. A detailed survey of various other stereo algorithms is found in [15].

Most algorithms differ from each other in the way they define primitives to be matched, their assumptions about the scene, and their incorporation of a priori knowledge. The assumptions on which our approach is based include constraints on continuity, uniqueness, the disparity gradient, and epipolar geometry. The stereo system is first calibrated and the parameters are recorded, thus ensuring that the horizontal scan line in the two images corresponds to the same scan line in 3D space. This reduces the matching process to a line-by-line operation, where line attributes depend on their positions and other constraints that must be satisfied. The vertical disparity can thus be neglected in further consideration. Although such solutions have been proposed, they continue to be an issue today due to the complexity of finding the corresponding points or objects in the two images. Various techniques that have been used include dynamic programming [4] and relaxation methods [6, 27, 34].

5.1.2 Our algorithm

We employ a feature-based approach to depth recovery and use detected line segments as features. The extracted line segments are grouped according to their most likely 3D orientation. The stereo matching problem is then greatly simplified. We consider only the lines oriented in the vertical and

horizontal direction. The second horizontal direction going into the image plane is ignored, as no definite starting and ending point is known and therefore it will give inaccurate 3D information. Further, we are not interested in precise depth information from horizontal lines, but need only a guideline in making the transition in motion direction. Correspondence is the most important stage in the process of stereo computation. Given the two images, correspondence must be established between detected features that are the projections of the same physical feature in the 3D scene. The imaging geometry creates distinct stereo paradigms. The search procedures are governed by the projection geometry of the imaging system and are expressed in the terms of epipolar constraints. Various local properties of the features must be matched in order to achieve a reasonable accuracy and success in the local matching process. The global consistency of these local matches is then tested by further constraints.

By restricting the matching process between detected line segments, we obtain coarse features that are also easier to match. As the line segments are grouped according to their orientations, the correspondence problem is further simplified. The correspondence between lines in the two images is achieved by associating weights to probable matches, which are calculated based on the disparity estimate, edge intensity value, length estimates, and the disparity gradient value. The general paradigm for depth recovery is common to many implementations, but ours is designed to be fast and efficient without loss of robustness.

Estimating the disparity value prior to the matching process provides more efficient feature matching by limiting the search area to only a certain section of the image. In order to achieve this, the baseline length of the lenses is calculated from the hardware setup and the lens calibration. The maximum search area in the stereo algorithm, which is the maximum disparity that can be encountered, is calculated as

$$disp_{max} = D.f/Z_{min}, \tag{22}$$

where D is the calculated baseline, f is the focal length, and Z_{min} is the closest possible depth that can be detected. The search area is further restricted based on the epipolar line constraint. As the orientation of each line is known, we can isolate the search to lines of similar orientation for the purpose of stereo matching. We use the method of iterative search in our stereo algorithm. Based on the value of maximum disparity, the search is iterated until a match is established. Knowing the endpoints, we can easily calculate the length of each line. It is very likely that lines in both the left and right images will have almost the same length. Thus, the search is also weighted according to lengths recorded for each match. The intensity value around the edge point of the lines is also used to associate weights to each corresponding match. An eight neighborhood of pixels around the edge point of the line is chosen and a summed intensity value is calculated. This is done as

$$Intensity_{i,j} = \sum_{i}^{n}\sum_{j}^{n} a_{ij} \ , \tag{23}$$

where a_{ij} is the edge pixel. This process is repeated for the edges of lines in the other image. A value is associated with the absolute difference of the intensity values between the two lines as

$$I_i = \|l_i - r_i\| \ , \tag{24}$$

where I_i is the intensity value, l_i and r_i are the edge intensities of the left and right line, respectively. Weight is associated to the matched pair according to the absolute intensity difference value and the minimum valued pair is chosen as a probable match. This is expressed as

$$M_i = arg\ min \sum_{i}^{n}\sum_{j}^{m} I_i.W_j \ , \tag{25}$$

where M_i is the weighted probable match and W_j are the weights associated with the intensity values. This process is repeated for each line in the database and weights are associated with each match. The final criterion for associating weights is based on the disparity gradient between two lines. Of the probable matches, two lines near to each other are selected, and one edge of each line is taken into account for each iteration. The disparity gradient value is then calculated based on the following equations. A_l, B_l, A_r, and B_r are the edge points of the selected lines, where the coordinates of each point is given as

$$A_l = (a_{xl}, a_{yl}); A_r = (a_{xr}, a_{yr}) \ , \tag{26}$$

$$B_l = (b_{xl}, b_{yl}); B_r = (b_{xr}, b_{yr}) \ . \tag{27}$$

The average coordinates of the two edge points between the two lines is given by

$$A = (a_{xl} + a_{xr})/2, (a_{yl} + a_{yr})/2 \ , \tag{28}$$

$$B = (b_{xl} + b_{xr})/2, (b_{yl} + b_{yr})/2 \ . \tag{29}$$

Now a separation value is calculated according to

$$S(A, B) = \sqrt{X^2 + Y^2} \ , \tag{30}$$

where

$$X^2 = (\frac{a_{xl} + a_{xr}}{2} - \frac{b_{xl} + b_{xr}}{2})^2 \ , \tag{31}$$

$$Y^2 = (\frac{a_{yl} + a_{yr}}{2} - \frac{b_{yl} + b_{yr}}{2})^2 \ . \tag{32}$$

The disparity gradient is calculated as the ratio of the disparity between the two edge points and the calculated separation value. This is given by

$$Grad_{disp} = \frac{X_l - X_r}{S(A, B)} \leq 1 \ . \tag{33}$$

The match with the minimum value is associated with the greatest weight.

The overall criterion for a match is set by examining the sum of the weights associated with each probable match. The match with the highest weight is then chosen as the correct match. The overall aim of the algorithm is to minimize the difference in the endpoints of the line in the left and right images and then reinforce the match by minimizing the difference in the length of each line segment, edge gradient, and disparity gradient. Thus, the approach of prediction and recursive verification of the hypothesis is used [2]. We have found that this procedure results in 98% true matches. Based

on the matches, we can then successfully calculate the depth of each segment and construct a map.

The robot moves by taking the stereo image pair at each step and estimates the 3D scene. While estimating the depth of various features and the spatial locations, it is necessary to consider the uncertainty in the position of the robot and the sensor data. We represent the uncertainty by a multivariate normal distribution with a mean vector $\boldsymbol{\mu}$ and covariance matrix $\sum$. The probability that a sensed point z is at $\mathbf{p}$ is then given by

$$f_z(\mathbf{p}) = \frac{exp[-\frac{1}{2}(\mathbf{p}-\boldsymbol{\mu})^T \sum_z^{-1}(\mathbf{p}-\boldsymbol{\mu})]}{2\pi\sqrt{|\sum_z|}} \ . \tag{34}$$

Given the motion of the robot, we can represent the local coordinate system with the robot in the center. The frames between motion intervals can be related by simple transformations from previous frames. Thus, all features in the 3D map can be represented by equi-probable ellipses which are the density contours of the multivariate normal distribution. Now we can determine the stereo uncertainty in estimating the 3D structure of the environment. In our stereo algorithm, we determine the location of vertical edges in the scene. Their 3D location can be associated with the normal distribution which describes the position uncertainty of the projected vertical point in 2D. As seen in Fig. 3, the focal length of the cameras f, and the baseline distance D between the cameras is known. The location of the edge point as projected on the image plane x_l and x_r are measured and associated with a normal distribution with a mean μ_{x_i} and variance σ_{x_i}. The location of the point in space is then given by

$$\mathbf{p} = \begin{pmatrix} F(x_l, x_r) \\ G(x_l, x_r) \end{pmatrix} = \begin{pmatrix} \frac{\frac{D}{2}(x_l+x_r)}{x_r-x_l} \\ f - \frac{Df}{x_r-x_l} \end{pmatrix}. \tag{35}$$

As this is a nonlinear function of x_l and x_r, $\mathbf{p}$ will not be normally distributed, but when σ_{x_i} is small, it is possible to linearize the functions F and G about the mean of x_l and x_r and then $\mathbf{p}$ will be a Gaussian. The functions F and G are linearized by taking the Taylor series expansion and then the points x and y can be given by

$$\begin{aligned} x &= F(\mu_{x_l}, \mu_{x_r}) + \frac{\partial F(x_l, x_r)}{\partial x_l} |_{x_l=\mu_{x_l}, x_r=\mu_{x_r}} (x_l - \mu_{x_l}) \\ &\quad + \frac{\partial F(x_l, x_r)}{\partial x_r} |_{x_l=\mu_{x_l}, x_r=\mu_{x_r}} (x_r - \mu_{x_r}) + \ldots \end{aligned} \tag{36}$$

$$\begin{aligned} &\approx \frac{D}{2}\frac{\mu_{x_l}+\mu_{x_r}}{\mu_{x_r}-\mu_{x_l}} + \frac{D\mu_{x_r}}{\mu_{x_r}-\mu_{x_l}}^2 (x_l - \mu_{x_l}) \\ &\quad - \frac{D\mu_{x_l}}{\mu_{x_r}-\mu_{x_l}}^2 (x_r - \mu_{x_r}) \ . \end{aligned} \tag{37}$$

Similarly, it can be shown that

$$\begin{aligned} y &\approx f - \frac{Df}{\mu_{x_r}-\mu_{x_l}} - \frac{Df}{\mu_{x_r}-\mu_{x_l}}^2 (x_l - \mu_{x_l}) \\ &\quad + \frac{Df}{\mu_{x_r}-\mu_{x_l}}^2 (x_r - \mu_{x_r}) \ . \end{aligned} \tag{38}$$

The Jacobian matrix can be now written as

$$\begin{aligned} J &= \begin{pmatrix} \frac{\partial x}{\partial x_l} & \frac{\partial x}{\partial x_r} \\ \frac{\partial y}{\partial x_l} & \frac{\partial y}{\partial x_r} \end{pmatrix} \\ &= \begin{pmatrix} \frac{D\mu_{x_r}}{(\mu_{x_r}-\mu_{x_l})^2} & \frac{-D\mu_{x_l}}{(\mu_{x_r}-\mu_{x_l})^2} \\ \frac{-Df}{(\mu_{x_r}-\mu_{x_l})^2} & \frac{Df}{(\mu_{x_r}-\mu_{x_l})^2} \end{pmatrix}. \end{aligned} \tag{39}$$

The covariance of point p is then given as

$$\sum_p = \begin{pmatrix} \sigma_x x & \sigma_x y \\ \sigma_x y & \sigma_y y \end{pmatrix} = J \begin{pmatrix} \sigma_{x_l}^2 & 0 \\ 0 & \sigma_{x_r}^2 \end{pmatrix} J^T \ . \tag{40}$$

The covariance matrix can be written as above with the off-diagonal terms equal to zero, because we assume that the locations of the edge points in the two images are independent. Further, we also assume that $\sigma_{x_l}^2 = \sigma_{x_r}^2 = \sigma^2$. Now the uncertainty of any point in space can be calculated and the precise spatial location in 3D is given by the mean and covariance of the triangulated point. Thus, the mean of the point $\boldsymbol{\mu}_\mathbf{p}$ is given as

$$\boldsymbol{\mu}_\mathbf{p} = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix} = \begin{pmatrix} F(\mu_{x_l}, \mu_{x_r}) \\ G(\mu_{x_l}, \mu_{x_r}) \end{pmatrix} \tag{41}$$

and the covariance $\sum_p$ is given by

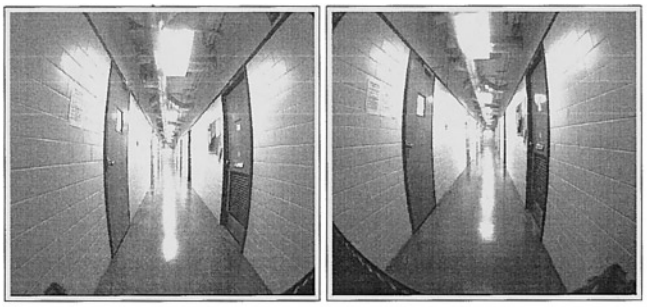
$$\begin{aligned} \sum_p &= \frac{D^2}{(\mu_{x_r}-\mu_{x_l})^4}\sigma^2 \\ &\quad \times \begin{pmatrix} \mu_{x_l}^2 + \mu_{x_r}^2 & -f(\mu_{x_l}+\mu_{x_r}) \\ -f(\mu_{x_l}+\mu_{x_r}) & 2f^2 \end{pmatrix}. \end{aligned} \tag{42}$$

Each reconstructed edge in 3D space can now be represented by an ellipse. The size of the ellipses grow bigger in length with respect to the distance from the robot. It is seen in our experiments that edges close to the cameras are determined with higher accuracy and can be represented by smaller ellipses. As the distance increases the uncertainty grows very fast and the lengths of the ellipses increase. A pair of stereo fish-eye images are shown in Fig. 8. The images are corrected for distortion and relevant segments are detected as shown in Fig. 9. Stereo correspondence is performed and the matched line segments from the left and right images are extracted as shown in Fig. 10. The 3D locations and the uncertainty associated with the estimates of the matches segments is shown in Fig. 11.

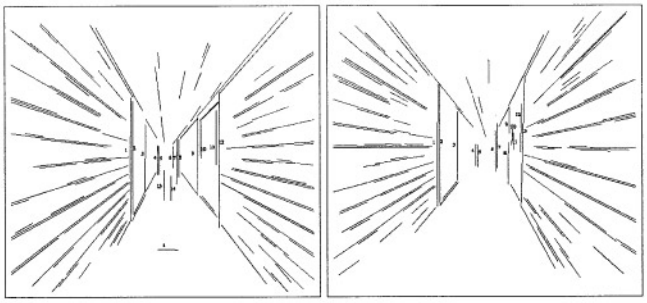
5.1.3 Fish-eye stereo accuracy

The algorithm was tested on several sets of images and the uncertainty in the 3D positions of the matched segments calculated. In order to determine the accuracy of the estimated depth for each line segment, we physically measured the depths of over 100 detected line segments in several image pairs. The average of these measurements and the average error are shown in Table 1.

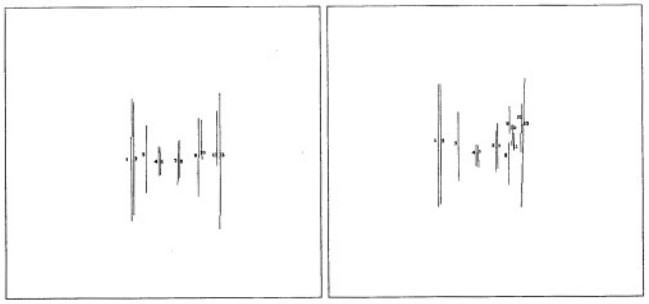
Results obtained by the fish-eye lens stereo system were also compared with those obtained by using wide-angle lenses to determine the advantage of using fish-eye lenses for autonomous navigation. A similar stereo setup was used with a pair of wide-angle lenses. A pair of images was acquired from the same positions as the fish-eye lenses. Once again,



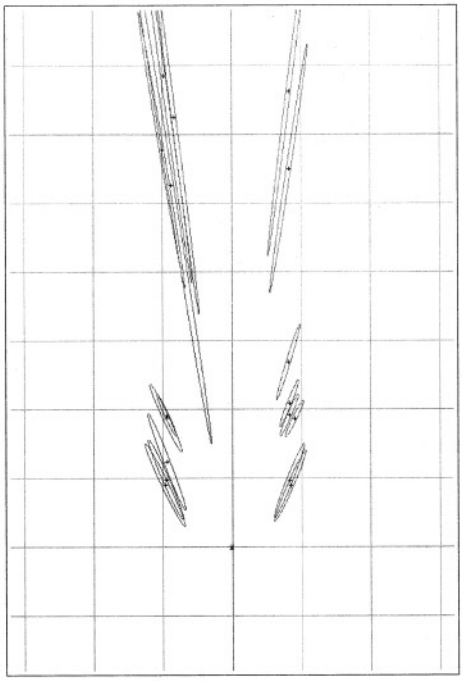
8



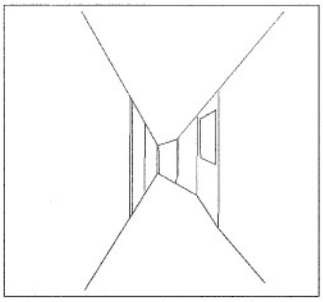
9



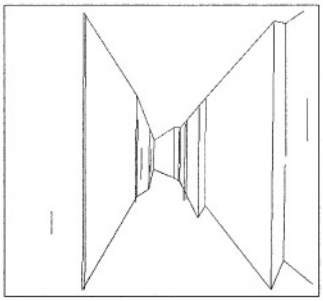
10



11



12a



12b

Fig. 8. Left and right image pair

Fig. 9. Segmented edge maps of the left and right image pair

Fig. 10. Matched vertical segments from the left and right edge maps

Fig. 11. Uncertainty in estimated depth

Fig. 12a,b. Spatial maps; **a** fish-eye and **b** wide-angle lens

Table 1. Estimated distance distribution

Detected line classification				
Segments:	Average distance estimation(mm)			
	#	Estimated	Measured	Error
Vertical	107	7549.6	8217.3	8.1 %
Horizontal	36	5427.9	6012.7	9.7 %
Total	143	7015.5	7512.6	6.6 %

Table 2. Lens error distribution

Lens classification			
Dist.:(m)	Average distance estimated		
	Parameters	Fish-eye	Wide-angle
	Segments	68	47
Less than	Estimated(mm)	1636.7	2539.7
4.3	Measured(mm)	1794.3	2859.5
	Error	8.7 %	11.2 %
	Segments	46	43
More than	Estimated(mm)	6516.96	6509.7
4.3	Measured(mm)	7490.16	7159.5
	Error	12.9 %	9.1 %

the significant lines were detected and the stereo correspondence established. The reconstructed models are compared in Fig. 12. For further details refer to [47].

The spatial information for the segments was calculated and compared to information previously obtained using the fish-eye lens stereo setup. Further, we compared the estimates of segments in two regions. We considered lines which were closer than 4.3 m and then those further away. The breakup in distance is based on the calculated maximum distance that a lens would see based on our setup, and is given by the equation

$$Z = f.D/d_{min} \ , \tag{43}$$

where f is the focal length, D is the baseline distance, and d_{min} is the minimum disparity observed in the image pair. The results of the comparison are shown in Table 2. We have found that lines closer to the lens can be estimated with higher accuracy by using the fish-eye lens, while lines further away are better estimated with the wide-angle lens.

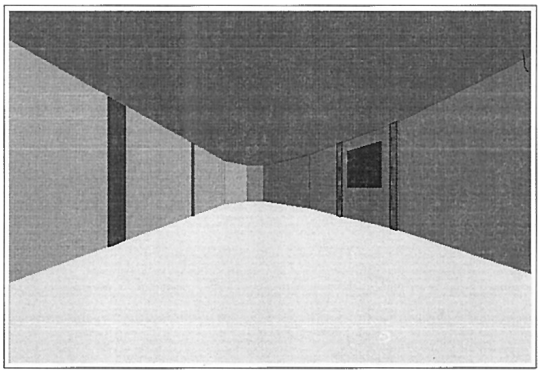


Fig. 13. The 3D depth map as estimated from a pair of stereo images

5.2 Depth recovery and scene representation

A projected line in the 2D image represents a plane in 3D. We consider the two endpoints of each matched line, and determine its 3D location. Each 3D point is represented by a vector $(x, y, z)^T$ and a covariance matrix $cov(x, y, z)$. Representing a 3D line in the parametric form with parameters (a, b, c), the 3D plane can then be represented by

$$ax + by + cz = d \ , \tag{44}$$

where the uncertainty is captured by $cov(a, b, c)$ and $var(d)$. Thus, a plane is defined by each endpoint of a matched 2D line. Two planes corresponding to the respective endpoint are represented according to (44). Then, the intersecting line gives the 3D position of the line. If the two planes are represented with parameters (a_1, b_1, c_1), and (a_2, b_2, c_2), then the 3D line parameters (a, b, c) can be given by

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} \times \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} \ , \tag{45}$$

where $\times$ is the cross-product. Figure 13 shows the 3D depth estimated from a pair of stereo images.

Having determined the 3D positions of detected line segments, it is necessary to update their locations once the robot has moved. The locations of various segments have shifted and their new locations need to be identified. Matching has to be performed between the new observations with the existing 3D representation of segments from an earlier image. With an existing 3D representation and an approximate estimate of the camera motion, it is possible to predict the location of a known edge in the new image. In the event of a segment being observed for the first time, the 3D location can be calculated by stereo correspondence and its 3D representation updated to the existing segments. The uncertainty for the new segment can be computed and the search space for the respective segment in new images can be adjusted appropriately. In the matching process, the estimated 3D orientation of the 2D segments is used by restricting the matching between similar orientations. This facilitates a faster and safer matching process. The order constraint is also implemented in order to eliminate poor matches. This constraint states that the order of line segments does not change from one image to the next. In cases with high angular motion, this constraint may not hold and is relaxed by reordering the segments based on the projections of the 3D segments. As the 3D orientation of segments is known a priori, the ordering is unique and simple. They are simply ordered according to increasing distances.

Establishing a successful match between an existing 3D segment with the new segments, requires an elimination process which disqualifies matches based on several measures. The segments are represented by a pair of their 2D and 3D representations and the distance between the 3D segment and the camera plane is calculated. For a possible match, the bounds for the distance are set between 0.5 m and 100 m. This information is useful for the following observations of the same 3D segment, as it allows for a bound on the search space within the new image. Therefore, for each segment appearing in the new image, the 2D lines must lie within finite bounds. This interval is calculated once for each expected segment. All segment observations not in the appropriate interval are rejected as matches. This ensures that segments are not behind the camera and not too much in front. We also consider the contrast of the corresponding 2D segment as a criterion for a possible match. Generally, the contrast or mean intensity gradient magnitude along the segment remains the same in image sequences, but could change due to drastic motion. To compensate for such motion, the average intensity and the contrast along each side of the segment is computed at a distance of one pixel from the edge. For a possible match, both the measures should be within 5% at least on one side of the segment. Furthermore, there should also be significant overlap between the observed and the existing segments. The 3D endpoints of the segment are projected to compute the 2D endpoints and are compared with the endpoints of the observed 2D segment. The ratio of the overlap is computed and, if the value falls below 0.8, the segment is rejected as a possible match. Only one-to-one matches are allowed and the matches are verified by the motion of the robot as recorded from the odometer readings.

5.3 Scene modeling

Visual navigation in indoor structured environments results in the representation of many interesting objects and features by planar patches bounded by linear edges. It is seen that an environment such as a corridor is mainly composed of linear edges with particular orientations in 3D. The linear edges are boundaries of opaque planar patches, such as the floor, ceiling, walls, etc. Repeated estimation and updates of the depth map via correspondence of the linear edges at each step allows the robot to make decisions regarding the navigable path. As the estimated depth has lower uncertainty close to the robot, it is possible to navigate in narrow environments. The robot is allowed to make a translation of 1.0 m at each time step and it is decreased to 0.5 m if a turn has to be made. To evaluate the possibility of navigation, the robot relies on cross-matches between detected segments. By considering segments closest to the approximated vanishing point, the depth to the segment is estimated. If this segment is more than 5.0 m away, the robot will decide to make a pure translation, along with correcting its heading. If the estimated space is below a certain threshold, the horizontal segments are considered and the depth furthest from the heading is es-

timated. This provides for initial motion direction, which is later updated according to the next image pair. The process is repeated and the robot can successfully navigate in indoor environments [46, 48, 49]. Finally, the depth maps estimated from the sequence of images are integrated to represent the model of the environment. By combining the 3D segments and assuming planar patches between them, a CAD model representation of the hallway may be constructed.

Knowing the 3D segment representations of the robot's environment, the CAD model can be generated by considering isolated segments. As the model construction is based on the segments with an uncertainty measure, it is possible that the resulting model is imperfect. The model will thus suffer from these shortcomings. Therefore, while generating the model, certain guidelines are applied.

1. Only well-observed, accurate 3D segments are considered.
2. A planar surface is hypothesized between any two parallel segments.
3. The planar surface is not considered if any other segment can be seen through it, or if the surface is incompatible with already established ones.

Furthermore, only vertical segments are considered as

- In a typical corridor scene, over 90 % of the reconstructed edges are vertical.
- Vertical edges are less likely to be partly hidden
- There could be conflicting information if more than one orientation was used.
- Floor plans can be constructed with just vertical edges.
- Most robot navigation schemes only use vertical edges.
- The algorithmic complexity of dealing with surfaces in full 3D is much greater, yet the benefits in practical situations are minimal.

The consequences of this is that ceilings with different heights, and uneven ground surfaces cannot be modeled.

While constructing a CAD model, the selected edge segments have to be relevant for representing a floor plan. The vertical segments selected have to be between an altitude of 0 and 2 m. This altitude criterion can be changed according to the minimum height in the environment. All segments should also have a minimum length to be considered significant. Each segment should also have a minimum number of observations and should not have a large uncertainty value associated with its location as calculated from the covariance matrix. A graph of these vertical edges is constructed which links the segments which have a chance of being connected by a planar surface in the 3D scene. The surface is defined as an opaque vertical rectangle, bounded by the floor, the ceiling, and the extension of the two vertical segments. For efficiency, this process is done in a 2D floor plan. In 2D, the segments become points, and surfaces become line segments. A partially connected graph is chosen to represent the links. The vertical segments are linked if the sum of the unsigned differences in x and y coordinates is less than 10 m apart. This measure was chosen because it favors surfaces that are aligned with the axes of rectangular buildings.

It is imperative that surfaces not be incorporated between segments such that vertical segments located beyond the surface are eliminated. Thus, surfaces through which another vertical segment was observed must be removed. For each combination of a surface hypothesis, a vertical segment, and a robot position from which this segment was observed, the algorithm checks whether the ray connecting the robot and the segment intersects the surface. If it does, the surface is transparent, and it is therefore not a valid hypothesis. The complexity of this search increases with the cube of the number of vertical segments and with the number of observations of each segment. Although this may seem high, several factors contribute to making the total processing time very short. First, the threshold on surface length eliminates a lot of unlikely hypotheses for large buildings: asymptotically, it makes the complexity follow the square of the number of segments instead of the cube. Second, the average number of observations for each segment seen in our experiments is well below 10. Finally, the CAD model needs to be built only once, after the robot has returned to base. The processing time was found experimentally to be negligible for most of the scenes. At the end of this processing step, the vertical segments can be connected by surfaces to any number of other vertical segments.

The other case for incorrect scene models is where a surface intersects with any other surface. The way to remove such surfaces is similar to the algorithm where surfaces intersected by an observation ray were eliminated. In such an event, the graph still contains some unnecessary surfaces. To remove such artifacts, the algorithm visits each vertical segment, starting from one of the camera positions. When going from one segment to the next, the surface that corresponds to the rightmost turn is chosen. After each vertical edge has been visited, the surfaces that were not used are eliminated. This algorithm is very efficient. However, it still does not eliminate every superfluous plane in the graph. Although this algorithm would also eliminate intersecting surfaces, this cannot be guaranteed unless we start from a fully connected graph. Another possible approach to further eliminate superfluous planes is to use hidden-line removal algorithms. This approach is less efficient but more robust than the previously proposed algorithm. In the present implementation, no superfluous surface removal is performed beyond the elimination of intersecting planes. The decision is left to the CAD operator, who can more easily delete extra surfaces than add omitted ones.

An algorithm estimates a uniform ceiling height using segments horizontal in 3D. It returns either the minimum, maximum, or average height of horizontal segments lying in a range of altitudes corresponding to most ceilings. The height is used by the next algorithm to extrude the floor plan into a 3D model. Of course, it is also possible to include the horizontal segments as 3D lines in the final CAD model and let a human operator connect them with 3D surfaces. An important feature of this approach is the creation of a description readable by most commercial CAD packages. The DXF file format was chosen for this reason. It is the native format of AutoCAD (of AutoDesk), and it is read by many other packages. DXF files are transferred to an Apple Macintosh and visualized with Virtus WalkThrough. This package lets users interactively explore a CAD model. Some difficult architectural scenes cannot be modeled accurately by the robot. If an edge is never seen by the robot, it would be difficult to represent that within a model generated from

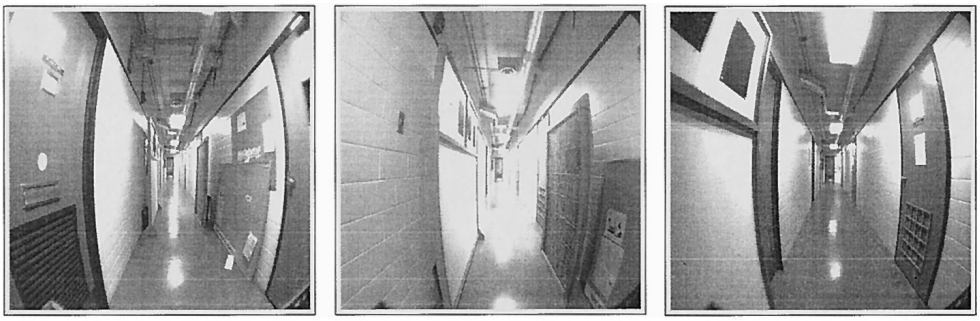


Fig. 14a–c. Three successive images from a typical sequence while navigating through a narrow passage

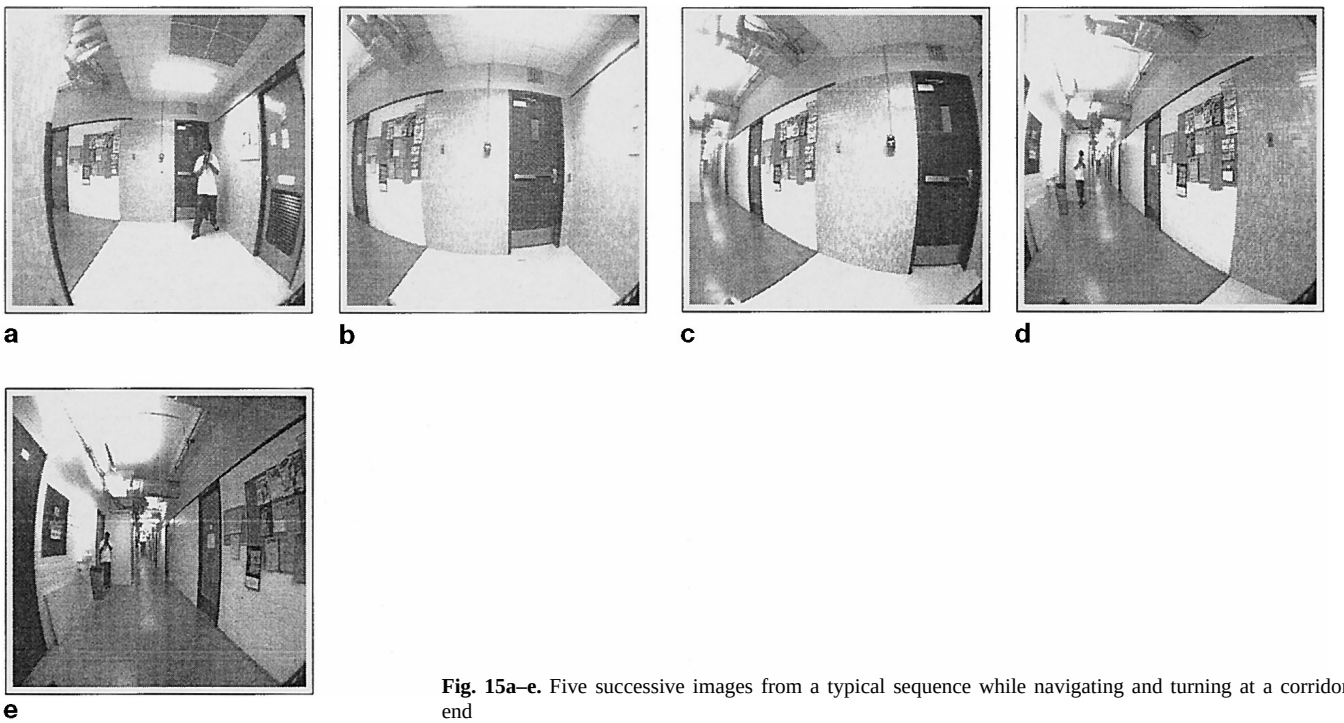


Fig. 15a–e. Five successive images from a typical sequence while navigating and turning at a corridor end

rest of the observations. One could imagine heuristics to model unseen elements; however, that approach has several drawbacks:

- A human operator correcting the CAD model would not easily know how planar surfaces have been generated. It would be better for him to see an obvious error than to see incorrect guesses.
- It is unclear how to deal with imperfectly aligned segments that are actually part of a wall.
- In many cases, no type of heuristic guessing can give the right answer.

6 Experimental results

6.1 Platform

The system has been implemented on *RoboTex*, the TRC-Labmate-based mobile robot [32]. *RoboTex* is a 1.5 m tall, tetherless mobile robot, weighing about 150 kg. It is used as an experimentation platform to demonstrate and test the vision algorithms presented in this paper. The robot subsystems comprise (1) a TRC Labmate base and rigid metal frame to support the equipment, (2) a fast, on-board HP-UX 735 UNIX workstation to digitize video images and control the robot, (3) a camera and digitizer, (4) an I/O system, (5) power supplies which enable completely autonomous operation, and (6) an off-board computing option, remaining from an earlier version of RoboTex. They are further detailed in [32].

6.2 Results

Numerous runs were made to qualitatively estimate the accuracy of the navigation algorithm. *RoboTex* was able to navigate through narrow passages and make rotations at corridor ends. Figure 14 shows three frames from a typical real corridor scene which is approximately 72 inches in width, and has been narrowed to 35 inches. These images are as seen by the robot during navigation. Figure 15 shows five frames as seen by the robot as it navigates around the corridor end and

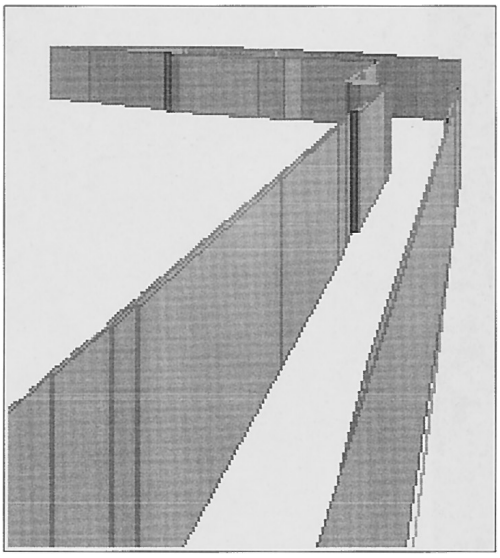


Fig. 16. CAD model of hallway by visual navigation using stereo fish-eye lenses

makes a rotation of 90 degrees. These images show qualitatively the effectiveness of the algorithm. The accuracy of the calculated 3D estimates can be found in [47].

The reconstructed model of the 3D scene is shown is Fig. 16.

An indication of the real-time performance of this system can be obtained by computing the time taken from acquiring the images to computing the motion. The following steps occur during this time:

1. Acquire images from stereo fish-eye lenses.
2. Undistort the images.
3. Segment and extract features.
4. Compute stereo correspondence and 3D estimate.
5. Calculate robot pose.
6. Make motion decision.

On a PA-RISC-based HP-735 workstation running at 99MHz, the time taken to perform the above is approximately 12 s.

7 Conclusion

In this paper we have presented a system for autonomous mobile robot navigation based on a stereo pair of fish-eye lenses that is capable of navigating through narrow corridors, making rotations at corridor ends and reconstructing 3D scene models of the navigated environment. The system has been implemented for navigation in a man-made environment where no a priori map is available. Using fish-eye lenses, we are able to estimate 3D information at close range to the robot and to sense the environment in more detail than is possible by using conventional lenses. The inherent distortion seen in fish-eye lens images is corrected and line segments relevant in an indoor scene are extracted. The correspondence procedure is simplified by using a specialized feature extractor and grouping lines according to their most likely 3D orientation. The robot pose is estimated using the 3D information and the robot's odometry is corrected by a vision-based algorithm. The system is implemented and the robot successfully navigates through passages with clearance of 4 inches on either side. The robot also makes rotations of 90 degrees at corridor ends. The algorithm for generating a CAD model of the navigated environment is also introduced and the result is presented for a corridor environment.

Acknowledgements. This research was supported in part by the Texas Advanced Technology Program Grant ATP-442 and in part by the Army Research Office under contract DAAH-04-94-G-0417.

References

1. Ayache N, Faugeras OD (1989) Maintaining representations of the environment of a mobile robot. IEEE Trans Robotics and Automation 5(6):804–819
2. Ayache N, Faverjon B (1985) A fast stereo vision matcher based on prediction and recursive verification of hypotheses. In: Proc. 3rd Workshop on Computer Vision: Representation and Control, pp 27–37
3. Ayache N, Faverjon B (1987) Efficient registration of stereo images by matching graph description of edge segments. Int. J. Comput. Vision 1:107–131
4. Baker HH, Binford TO (1981) Depth from edge and intensity based stereo. In: Proc. 7th Int. Joint Conf. on Artificial Intelligence, Vancouver, Canada, pp 631–636
5. Barnard ST (1988) Stochastic stereo matching over scale. In: Proc. DARPA Image Understanding Workshop, Cambridge, MA, pp 769–778
6. Barnard ST, Thompson WB (1980) Disparity analysis of images. IEEE Trans Pattern Analy Mach Intell 2:333–340
7. Bouguet J-Y, Perona P (1995) Visual navigation using a single camera. In: Proc. International Conference on Computer Vision, pp 645–652
8. Chang Y, Lebègue X, Aggarwal JK (1993) Calibrating a mobile camera's parameters. Pattern Recognition 26(1):75–88
9. Chatila R, Laumond J-P (1985) Position referencing and consistent world modeling for mobile robots. In: Proc. IEEE Int. Conf. Robotics and Automation, St. Louis, pp 138–145
10. Chattergy A 1985): Some heuristics for the navigation of a robot. Robotics Research 4(1):59–66
11. Cox IJ, Wilfong GT (1990) Autonomous Robot Vehicles. Springer, Berlin Heidelberg New York
12. Crowley J (1985) Navigation of an intelligent mobile robot. IEEE J Robotics Automation (1):31–34
13. Dalmia AK, Trivedi MM (1996) Depth extraction using a single moving camera: An integration of depth from motion and depth from stereo. Mach Vision Appl 9:43–55
14. Robert de Saint Vincent A (1986) A 3D perception system for the mobile robot hilare. In: Proc. of the IEEE Int. Conf. Robotics and Automation, San Francisco, Calif., pp 1105–1111
15. Umesh Dhond R, Aggarwal JK (1989) Structure from stereo–a review. IEEE Trans Syst Man Cybernetics 19:1489–1510
16. Faugeras OD, Ayache N (1987) Building, registering and fusing noisy visual maps. In: Proc. First Int. Conf. Computer Vision, London, UK, pp 73–82
17. Faugeras OD, Ayache N, Faverjon B (1986) Building visual maps by combining noisy stereo measurements. In: Proc. of the IEEE Int. Conf. Robotics and Automation, San Francisco, Calif., pp 1433–1438
18. Gennery DB (1980) Object detection and measurement using stereo vision. In: Proc. ARPA Image Understanding Workshop, College Park, MD, pp 161–167
19. Giralt G, Sobek R, Chatila R (1979) A multi-level planning and navigation system for a mobile robot; a first approach to hilare. In: Proc. Sixth International Joint Conference on Artificial Intelligence, pp 335–337
20. Grimson WEL (1985) Computational experiments with a feature based stereo algorithm. IEEE Trans Pattern Anal Mach Intell 7(2):17–34
21. Hannah MJ (1980) Bootstrap stereo. In: Proc. ARPA Image Understanding Workshop, College Park, Md, pp 201–208

22. Hannah MJ (1985) Sri's baseline stereo system. In: Proc. DARPA Image Understanding Workshop, Miami, FL, pp 149–155
23. Ishiguro H, Yamamoto M, Tsuji S (1992) Omni-directional stereo. IEEE Trans Pattern Anal Mach Intell 14(2):257–262
24. Iyengar SS, Elfes A (1991) Autonomous Mobile Robots. IEEE Computer Society Press, Los Alamitos, Calif.
25. Kak AC, Andress KM, Lopez-Abadia C, Carroll MS (1990) Hierarchical evidence accumulation in the PSEIKI system and experiments in model-driven mobile robot navigation. In: Uncertainty in Artificial Intelligence (Vol. 5), Elsevier, North-Holland, pp 353–369
26. Kim D, Nevatia R (1993) Indoor navigation without a specific map. In: Proc. Int. Conf. Intelligent Autonomous Systems, Pittsburgh, PA, pp 268–277
27. Kim YC, Aggarwal JK (1987) Positioning three-dimensional objects using stereo images. IEEE J Robotics Automation 3:361–373
28. Kosaka A, Kak AC Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. Comput Vision Graphics Image Process Image Understanding 56(3):271–329
29. Lebègue X, Aggarwal JK (1992) Detecting 3-D parallel lines for perceptual organization. In: Proc. Second European Conf. on Computer Vision, pp 720–724, Santa Margherita Ligure, Italy, Springer, Berlin Heidelberg New York
30. Lebègue X, Aggarwal JK (1992) Extraction and interpretation of semantically significant line segments for a mobile robot. In: Proc. IEEE Int. Conf. Robotics and Automation, pp 1778–1785, Nice, France
31. Lebègue X, Aggarwal JK (1992) A mobile robot for visual measurements in architectural applications. In: Proc. IAPR Workshop on Machine Vision Applications, pp 195–198, Tokyo, Japan
32. Lebègue X, Aggarwal JK (1993) Robotex: An autonomous mobile robot for precise surveying. In: Proc. Int. Conf. Intelligent Autonomous Systems, pp 460–469, Pittsburgh, PA
33. Lebègue X, Aggarwal JK (1994) Automatic creation of architectural CAD models. In: Proc. 2nd CAD-Based Vision Workshop, Seven Springs, PA
34. Marapane SB, Trivedi MM (1990) Edge segment based stereo analysis. SPIE Appl Artif Intell VIII 1293:140–151
35. Marr D, Poggio T (1979) A computational theory of human stereo vision. In: Proc. Royal Soc London B204:301–328
36. Matthies LH, Szeliski R, Kanade T (1989) Kalman filter-based algorithms for estimating depth from image sequences. Int J Comput Vision 3:209–236
37. Medioni G, Nevatia R (1985) Segment-based stereo matching. Comput Vision Graphics Image Process 32:2–18
38. Mohan R, Medioni G, Nevatia R (1989) Stereo error detection, correction, and evaluation. IEEE Trans Pattern Anal Mach Intell 11(2):113–120
39. Moravec HP (1977) Towards automatic visual obstacle avoidance. In: Proc. 5th Int. Joint Conf. Artificial Intelligence, p 584
40. Moravec HP Robot Rover Visual Navigation. UMI Research Press, Ann Arbor, Mich.
41. Moravec HP The stanford cart and cmu rover. Proc IEEE 71(7):872–884
42. Nevatia R, Babu K (1980) Linear feature extraction and description. Comput Vision Graphics Image Process 13:257–269
43. Olivieri P, Gatti M, Straforini M, Torre V (1992) A method for the 3D reconstruction of indoor scenes from mono cular images. In: Proc. Second European Conf. on Computer Vision, pp 696–700, Santa Margherita Ligure, Italy, Springer, Berlin Heidelberg New York
44. Parodi A (1985) Multi-goal real-time global path planning for an autonomous land vehicle using a high speed graph search processortion. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, St. Louis, Missouri, pp 161–167
45. Shah S, Aggarwal JK (1994) A simple calibration procedure for fish-eye (high distortion) lens camera. In: Proc. of Int. Conf. on Robotics and Automation, San Diego, Calif., pp 3422–3427
46. Shah S, Aggarwal JK (1995) Autonomous mobile robot navigation using fish-eye lenses. In: Proc. Third International Computer Science Conference, Hong Kong, pp 9–16
47. Shah S, Aggarwal JK (1995) Depth estimation using stereo fish-eye lenses. In: Proc of Int Conf on Image Processing, Austin, Texas, pp 740–744
48. Shah S, Aggarwal JK (1995) Modeling structured environments using robot vision. In: Proc of Third Asian Conf in Computer Vision, Singapore, pp 297–304
49. Shah S, Aggarwal JK (1995) Modeling structured environments using robot vision. Lecture Series in Computer Science: Recent Progress in Computer Vision, pp 113–128
50. Shah S, Aggarwal JK (1996) Intrinsic parameter calibration procedure for (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. Pattern Recognition 29(11):1775–1788
51. Straforini M, Coelho C, Campani M, Torre V (1992) The recovery and understanding of a line drawing from indoor scenes. IEEE Trans Pattern Analy Mach Intell 14(2):298–303

Shishir Shah received the B.S. degree in Mechanical Engineering in 1994, and the M.S. degree in Electrical Engineering in 1995 from The University of Texas at Austin. He is presently working toward the Ph.D. degree at the Computer and Vision Research Center, The University of Texas at Austin. His research interests include mobile robots, computer vision, scene analysis, and pattern recognition. Mr. Shah is a student member of the IEEE Computer Society.

J.K. Aggarwal has served on the faculty of The University of Texas at Austin College of Engineering since 1964 and is currently the Cullen Professor of Electrical and Computer Engineering and Director of the Computer and Vision Research Center. His research interests include computer vision, parallel processing of images, and pattern recognition. He has been a Fellow of IEEE since 1976. He received the Senior Research Award of the Americal Society of Engineering Education in 1992, and was recently named as the recipient of the 1996 Technical Achievement Award of the IEEE Computer Society. He is author or editor of 7 books and 31 book chapters; author of over 170 journal papers, as well as numerous proceedings papers and technical reports. He has served as Chairman of the IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence (1987–1989); Director of the NATO Advanced Research Workshop on Multisensor Fusion for Computer Vision, Grenoble, France (1989); Chairman of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1993), and President of the International Association for Pattern Recognition (1992–94). He currently serves as IEEE Computer Society representative to the IAPR.