ELSEVIER

Robotics and Autonomous Systems 26 (1999) 185–201

Robotics and Autonomous Systems

Robotic sensing for the partially sighted

Nicholas Molton [a,*], Stephen Se [a,1], Michael Brady [a,2], David Lee [b,3], Penny Probert [a,4]

[a] *Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK*

[b] *Department of Electrical and Electronic Engineering, University of Hertfordshire, College Lane, Hatfield, Herts AL10 9AB, UK*

Received 25 June 1998; accepted 30 August 1998

Abstract

Partial sightedness is a sensory disability which can to some extent be alleviated by artificial aids. Many of the sensory methods used in robotics can be applied in attempts to recapture some of the sensory information a partially sighted person has lost. This paper describes a device which uses sonar and stereo vision sensors for this task. The device is portable, and is worn by the user, giving them freedom of movement over kerbs, stairs and rough ground. Sensor motion during walking is measured using visual egomotion recovery and odometry, and has been modelled to allow compensation in the sensor readings. A ground position estimate is continually updated by scene ground-plane fitting, or from the walk-motion model, and is used to classify scene features as obstacles or parts of the ground. Methods for the robust reconstruction of image points and lines into scene features are developed. The recognition of world objects of exceptional significance to a mobile person – kerbs and stairs – is given particular attention. A user interface, which has undergone limited real world testing, is also described. Experimental results are presented for the various parts of the system.

Keywords: Partially sighted; Obstacle detection; Stereo vision; Stair-case detection

1. Introduction

Despite the potentially great benefit, many of the 40 million visually impaired people worldwide use no form of mobility aid in their everyday lives. The non-human aids that are most commonly adopted are the long cane (by about 50% in the UK) and guide dog (by about 4% in the UK). These have had limited acceptance, essentially because the long cane has a very limited range, and guide dogs are very expensive and difficult for a blind person to look after. In addition, neither the long cane, nor the guide dog give warning of obstacles at head height such as overhanging branches.

The most popular electronic mobility aids used today are those based on conversion of sonar information to an audible signal for the user to interpret [6,18,19]. These are not, however, commonly used as they require extensive training or provide limited information. An alternative approach is the mapping of images onto tactile arrays fixed to the skin [2], though this has proved problematic. Research groups are also looking at the possibility of connecting cameras straight into the human nervous system [3]. Our approach attempts to interpret the visual information and convert it to a higher level representation before sending it to the user.

* Corresponding author. E-mail: ndm@robots.ox.ac.uk

[1] E-mail: syss@robots.ox.ac.uk

[2] E-mail: jmb@robots.ox.ac.uk

[3] E-mail: d.c.lee@herts.ac.uk

[4] E-mail: pjp@robots.ox.ac.uk

0921-8890/99/$ – see front matter
PII: S0921-8890(98)00068-2

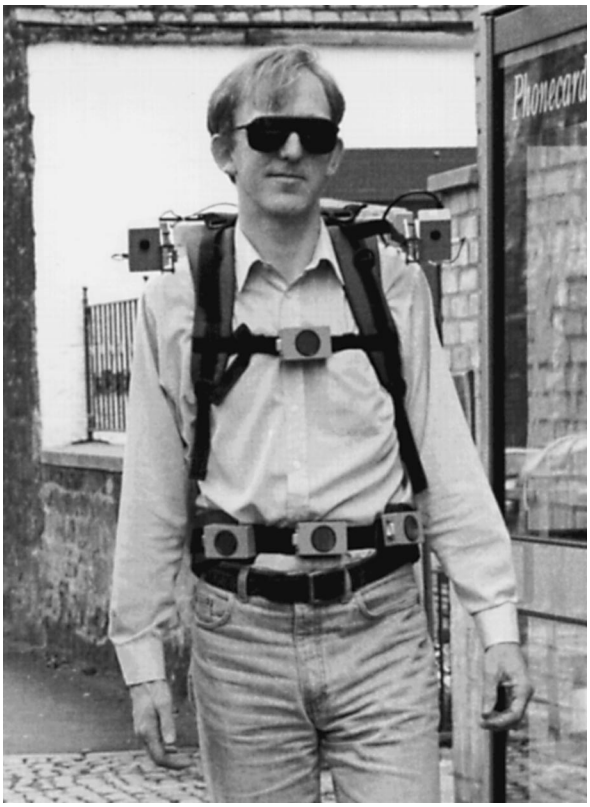

Fig. 1. The project prototype backpack

1.1. System design

It is important that the user should be free to walk naturally, rather than being pulled or abruptly stopped by an aid. This is achieved by having the user carry the device. They are also given a greater degree of freedom of movement, being able to negotiate stairs and hills, in principal even to run. Body mounting also gives the device the potential to be small and discrete.

The current project prototype is shown in Fig. 1. The system is based around a backpack which houses the system electronics. Three sonar sensors are mounted on the user's belt and one on the chest. These are driven and interpreted by a Motorola HC11 microcontroller. Rigid arms extend from the backpack over each shoulder, and onto each of these is mounted a calibrated greyscale camera. Camera orientation is adjustable to allow the cameras to be aligned. The cameras are interfaced to an image capture and processing board based on the TI C40 processor. Images can be captured in stereo-pairs (an image captured from the right and left cameras at the same time). Feedback to the user is through vibrating motors attached around the users belt.

The combination of sonar and vision was chosen because of the complementary nature of the sensors. Sonar measures range accurately but not direction, whereas stereo vision measures direction more accurately than range. Both devices will fail under certain circumstances but are unlikely to fail together: sonar is confused by reflections from the ground and multiple reflections, while vision shows unpredictable behaviour when mirrors and windows are viewed.

The project aim is for a device capable of detecting obstacles at least 10 cm high, at a range of up to 5 m.

1.2. Paper overview

The remainder of this paper describes ongoing work into the computer vision and sonar components of the system. More discussion is given to the vision issues as this is thought to be the more profitable research area.

A model of the movement experienced by sensors mounted on a walking person is developed in Section 2. Determination of ground position from image lines is then achieved in Section 3. The line features used can be classified as obstacle or ground features as a result of this. Section 4 then deals with the reconstruction and tracking of points in the world.

Detection of negotiable hazards, such as stairs and kerbs is discussed in Section 5. The detection of doorways and traversable paths has been studied in related work on this project [23]. The user interface was tested on a guide dog training course in conjunction with the sonar sensor part of the system. This is described in Section 6. The final part of the paper, Section 7, is a summary and discussion of the likely future direction of the work.

2. Walker egomotion measurement

For a proper understanding of the system as described, a model of the effect of mounting sensors onto a walking person was first developed. This is useful in two ways: firstly to provide more accurate

measurement of recent motion from uncertain motion measurements; and secondly as a predictor of the likely next movement.

A knowledge of recent motion is beneficial because it allows an estimate of the position of the ground to be made from previous ground observations, at times when the ground cannot be observed. Knowledge of the motion also facilitates its removal from the motion of independently moving objects in the world to reveal their underlying motion, which can then be more easily tracked. Movement prediction provides estimates of feature image location, which allows image features to be more easily matched through time.

2.1. Motion measurement

Motion of the project sensors can be measured using the sensors themselves, or independent odometry. The measurement of motion visually is discussed in Section 4, and in more detail in [15]. The current section describes the result of attempts to track rotational motion measured using a digital compass and inclinometer.

The gait of a walking person was first analysed in an orthopedic gait lab [8]. This demonstrated that the six components of motion are sinusoidal in nature. Measurements from the rotation sensors used (roll, pitch, and yaw), after filtering with a Butterworth filter, confirm this. These motion signals are analysed and tracked in real time using a combination of parameter extraction using wavelets, and an extended Kalman filter. This is discussed further in [14].

2.2. Motion modelling

The instantaneous parameters of a sinusoid approximation to each motion component are first extracted by convolution with Gabor wavelets. Convolution of the latest part of a signal with odd and even wavelets over a suitable range of frequencies gives a complex response for each frequency. The response of greatest magnitude defines the signal frequency, the corresponding phase of the response gives the signal phase, and signal amplitude can be calculated from the magnitude of the response.

This method of parameter extraction is reliable and accurate [7], however, because the wavelets are double sided, the parameters extracted characterise the signal

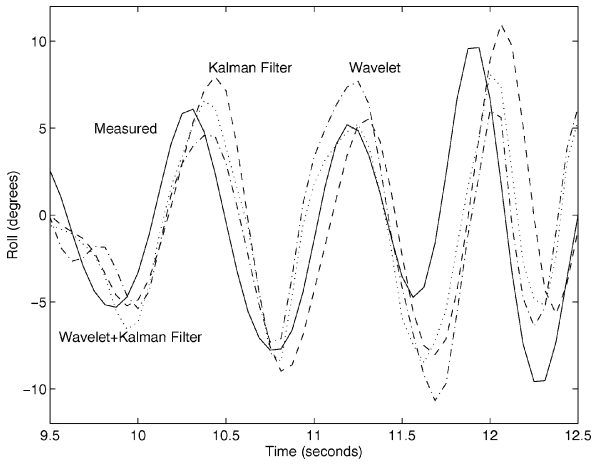

Fig. 2. A roll signal, and predictions of it made 0.25 s in advance.

$\psi/2$ from the latest reading – where ψ is the width of the wavelet used. As ψ must be of a certain size, the parameter estimates are never quite current.

For this reason, analysis continues with an iterated, extended Kalman Filter [1] initialised on the wavelet parameter estimates, and running from time $\psi/2$ to the last available reading. Each component of rotation is approximated by a single sine wave, described by

$$C_i = A_i \cos(\theta_i) + D_i. \tag{1}$$

It is assumed that frequency (f_i), offset (D_i), and amplitude (A_i) of each component are nominally fixed (allowed to vary slowly in the Kalman filter). Phase angle (θ_i) is updated at each step from f_i. Because the three frequency components are linked ($f_{\text{pitch}} = 2f_{\text{roll}} = 2f_{\text{yaw}}$) the number of filter parameters can be reduced from 12 to 10. The additional provision of a linear variation in offset for the yaw signal must also be given, to properly model smooth corner turning.

An example of prediction error for a roll measurement using this method is shown in Fig. 2. The example is particularly testing because it contains sudden variation in walking speed. The results for a Kalman filter, or wavelet analysis alone are shown for comparison. These are seen to perform slightly worse than the combined method.

State detectors are also used, for example to start the tracking process when walking is thought to have started.

3. Obstacle detection using edges

3.1. Ground plane obstacle detection

Ground plane obstacle detection (GPOD) using stereo disparity was first reported by Sandini [4] and subsequently refined by Mayhew [11] and by Li [9,10]. GPOD uses a pair of cameras to detect features which do not lie on the ground plane.

GPOD parameterises the ground plane using measurements of disparity, rather than the depths of world features. This improves robustness to calibration errors. It is a feature-based stereo algorithm. Images of the ground that contain line features but no obstacles are used to initialise the ground plane estimate.

GPOD works in image coordinates, and compares the disparity values in a new image pair with the expected ground plane disparity to detect differences. The measured disparity of an obstacle will be significantly larger than that which has been predicted for the ground. Vertical edges are detected using a Sobel detector, and stereo matching uses the PMF algorithm [16,17]. Special cases of the Sobel and PMF algorithms decrease the cycle time [9].

The ground plane disparity d varies linearly with cyclopean image plane position [10], i.e.

$$d = au + bv + c, \tag{2}$$

where (u, v) is the cyclopean image coordinate. In initialisation, a least-squares fit is used to estimate the *ground plane parameters* (a, b, c). The algorithm has been extended to sub-pixel accuracy by parabolic interpolation during Sobel edge detection and a better fit with lower residue is obtained.

3.2. Dynamic ground plane recalibration

For wheeled mobile robots moving over flat ground, there is no change in position of the ground plane relative to the cameras, and the cyclopean ground-plane disparity function is therefore fixed. However, when mounted on a person, the sensor movements make it impossible to use a one-time, fixed ground plane calibration in this application.

When many ground features can be seen, we use ground-plane fitting based on RANSAC [5] to estimate the ground-plane parameters. RANSAC takes all the image features (provided that there are sufficient ground plane features, and not all obstacle features lie on the same plane), fits the ground plane features and discards the obstacle features as outliers.

For RANSAC, the probability of a good sample (all inliers) [24] is given by

$$\Upsilon = 1 - (1 - (1 - \varepsilon)^p)^m, \tag{3}$$

where ε is the contamination fraction, p the size of the sample and m is the number of samples. The size of the sample in this case is 3, as we need three points to define a plane, and we assume that the percentage of contamination for a typical scene is 75%. If we wish to achieve a 99% probability of a good sample, the number of samples required is around 300.

The algorithm for *dynamic ground plane recalibration* (DGPR) is as follows [15,20].

Iteratively at each step, we recalibrate the ground plane as discussed above to determine the ground plane parameters for obstacle detection. Features found are partitioned into two types: ground plane features and obstacle features.

Track initiation, maintenance and termination is performed for all the features found. Obstacle feature information is then used to send alarms to the user about nearby frontal obstacles.

When there are not sufficient ground features to recalibrate the ground plane, we use the previous step's ground plane parameters and the camera motion estimate (from Section 2) to predict the ground plane parameters of the current frame. Since the motion estimate is difficult to obtain accurately over long periods [12], when there are enough features, recalibration with RANSAC will refit the ground plane to avoid accumulation of camera motion estimation errors.

3.3. Experimental results

We have carried out numerous experiments to determine the utility of dynamic recalibration. In the results presented here, a sequence of stereo images of a real outdoor scene was captured at 128×128 resolution. The environment is a tiled pavement with various obstacles. There was some camera motion between the images with translations up to 20 cm and rotations up to 5°, which cover the extreme case for human movement [8].

The same sequence of images was used to test both the DGPR algorithm and the RANSAC–DGPR.

Table 1
Comparison between GPOD and DGPR results

Step number	1	2	3	4
Total no. of obstacles in region	6	6	6	8
No. of obstacles detected by GPOD	4	0	0	0
No. of obstacles detected by DGPR	4	4	4	5

Obstacle objects are found from matching the obstacle edges using a heuristic edge matching technique [22]. A comparison between GPOD and DGPR in terms of the number of obstacles detected in the first four images is shown in Table 1. In these tests, the between-image movements were measured approximately by hand. From the second image onwards, the basic GPOD algorithm fails to detect any obstacles, but the DGPR algorithm is unaffected by the movements. The program does not run in real time on the C40 processor but will run at 2 Hz on a Pentium 166, and at 4 Hz on a state of the art 300 Hz machine.

4. Point reconstruction and tracking

The robust reconstruction of points as well as lines is considered because they occur more frequently than do lines, particularly in nature (for example, in a bush). The most important aim of the reconstruction is that grossly erroneous points must not be allowed to enter it. It is also desirable to have as many points as possible in the reconstruction. The method described in this section concentrates especially on the first of these aims. Any camera motion between frames is permitted, the motion need not be smooth, so long as there is an overlap in that seen in consecutive images. Any type of scene structure, and depth range, is permitted, as a full projective model of the scene is used. The method assumes scene rigidity, but can be extended to allow independently moving near-rigid objects.

4.1. Robust point structure and motion from stereo

Between the capture of successive image pairs four stages are performed. Stage A matches between points in a stereo-pair of images, stage B matches between one of these images and the corresponding image in the following stereo-pair (the latest available pair).

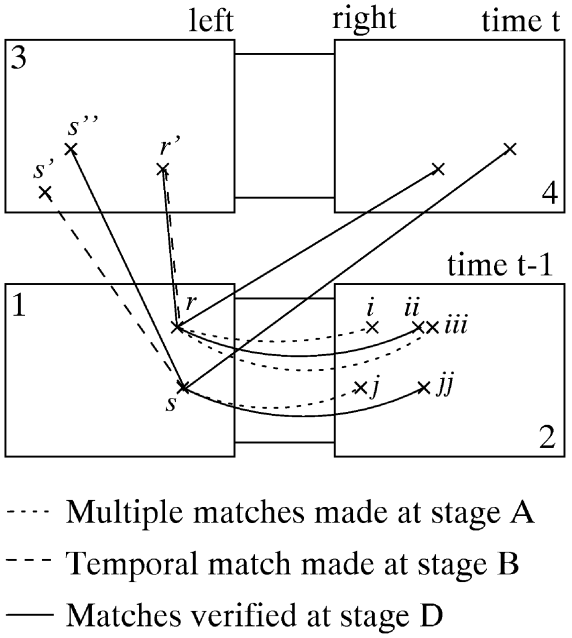

Fig. 3. Point matching between four image pairs.

Stage C estimates camera motion between the two stereo-pairs using the matches made so far. Stage D then verifies and corrects the matches made against the camera motion estimate and provides matches to the remaining image in the latest stereo-pair. For a more accurate motion estimation result iteration should be performed between stages C and D. The order after receiving a new image pair is actually BCDA.

Fig. 3 illustrates the processes occurring during this cycle, and is referred to in the text. The effect on the number of points reconstructed correctly and incorrectly at the end of stage D, of varying one of the process parameters, while the others are held fixed, is shown in Fig. 4.

4.1.1. Stage A: Stereo matching of unmatched points

For the latest stereo-pair of images, stereo matching (matching of points from one image to the other in the stereo-pair) of *new* points (those not matched by stage D at the previous iteration) is first performed. Features are matched by comparison of image intensity values in a small area around each corner. The comparison is invariant to intensity and contrast differences between the areas, and limited topological differences are allowed [13].

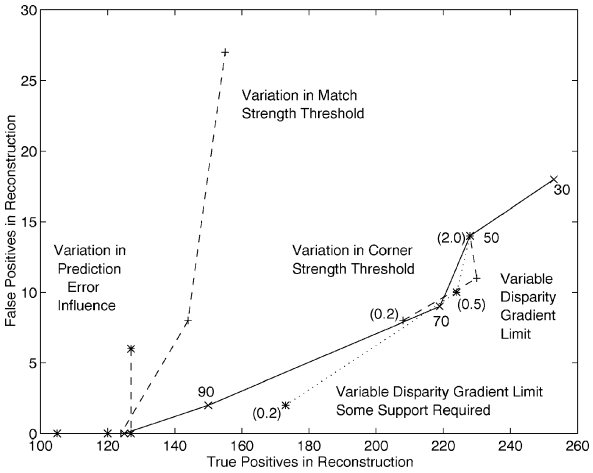

Fig. 4. The dependency of reconstruction accuracy and size on system parameters.

The strongest few matches to a left image point in the right image (a maximum of four matches) are recorded. Referring to Fig. 3, the last time this was done (at time $t-1$), r was matched to i, ii, and iii, and s to j and jj. The most important benefit in allowing the stereo match to have multiplicity is that *all* plausible matches to a point are retained until the true match can be reliably recognised. Ordinarily, the number of new points processed this way (a one-dimensional search) will be a small fraction of the total, most being matched by the more efficient verification method of stage D (a small area search).

4.1.2. Stage B: Temporal image matching

In this stage points are matched between consecutive left (or right) images, based again on image match correlation. A search is made for the match within an area of the point's previous position, or if a movement (gait) model such as that described in Section 2 is available, it is used to guide a more efficient search (compactness of the search area required, hence efficiency, is a function of the motion model accuracy). The matches made will be imperfect but provided camera motion between the two images is found to be sufficiently confident at the next stage (if the motion of a good number of points concur with it), the matches made now can be later ignored. In Fig. 3, matching is carried out between left images, r is matched to r', and s to s'. Additional robustness would be achieved if this stage were repeated between consecutive right images and the results combined.

4.1.3. Stage C: Initial estimate of motion

Matches established in the previous two stages, though crude in stage B, should contain enough information to allow robust estimation of the motion between the two stereo-pairs. This is achieved by random sampling of sets of three points (taking a random match where multiple stereo matches exist), calculation of the motion implied by the matches for this set, and a choice of motion which is consistent with the greatest number of the remaining point matches (a RANSAC method). This method, described further in [13,14], was found to be more robust than a method based on the fundamental matrix, which is degenerate under certain motions [24]. Any combination of rotation and translation, or no motion at all, is permitted. An example of a random three point set might be the point r (with matches $r-i$ and $r-r'$), the point s (with matches $s-jj$ and $s-s'$), and a third point, with one of its stereo matches randomly selected. The use of *multiple* hypothesised stereo matches is better than the use of a single best match because more correct matches will be available to provide support. It is however necessary to adjust the number of samples taken within RANSAC to allow for the extra outliers. Eq. (3) is modified to give

$$\text{Number of samples required} = \frac{\log(1-\Upsilon)}{\log(1-(\sigma_1/M)^3)}, \tag{4}$$

where Υ is the probability that a sample containing only inliers is chosen, M is the mean match hypothesis multiplicity at stage A (stereo matching), and σ_1 is an estimate of the proportion of inliers in the dataset (points matched correctly in stereo and through time) if each point were allowed only one stereo match. In practice this will be an *overestimate* of the number of samples required, which is satisfactory.

If a significant portion of the resulting matches are inliers (a figure of 20% is used) a correct motion hypothesis is assumed, otherwise stages B and C are repeated with a wider search area until a sufficiently supported motion is found.

4.1.4. Stage D: Verify and update matches

In this section an attempt is made to match points over the four images shown in Fig. 3: the previous stereo-pair (numbered 1 and 2), and the latest pair (numbered 3 and 4). In stage A an attempt has been made to define stereo matches between points in images 1 and 2, in many cases resulting in multiple matches for each point. In this stage we wish to ascertain which of these matches are reliable using the information content of the new image pair and the estimate of camera motion between the pairs.

Considering each hypothesised match in turn, a point position in image 1, a match disparity, and a match correlation α_{12} are known. From the position and disparity the point's location in space at the earlier time step can be determined, the motion estimate used to transform this location to the latest time step, and the new location projected to give predicted position in images 3 and 4. The point should appear close to these positions if the stereo match hypothesis is correct. A confined search for the point is conducted very near to the positions and the best correlation found to image 3 (α_{13}) and image 4 (α_{14}) is recorded.

A quantity, Ψ, is derived to rate the hypothesis based on the similarity of the matched image points and their accordance with the rigid motion. The rating is defined by the match correlations α_{12}, α_{13}, and α_{14}, and deviations from the implied motion:

$$\text{Match strength, } \Psi = \frac{\alpha_{12}\,\alpha_{13}\,\alpha_{14}}{d_L + d_R + \varepsilon_d} \tag{5}$$

The values d_L and d_R are the distances between the predicted and matched point positions (the deviations) in images 3 and 4 respectively. The constant ε_d, *prediction error influence*, balances the importance placed on visual similarity over rigidity in the reconstruction.

For a point to become verified its match strength Ψ must lie above a *match strength threshold*. Many of the multiple hypotheses for a point will be eliminated by this threshold but for some points more than one verified match may still exist. Two tests are performed in an attempt to isolate a single match for these. The first, a supremacy test accepts a hypothesis if its match strength is larger than any other by at least a factor t_s, the *supremacy threshold* (set to 5). The second test is concerned with stereo match hypotheses made which are very close together (image 2 positions within a *proximity threshold*). Sometimes when trying to match two features using image patches, because of pixelation and viewpoint induced deformation there may be more than one place around the true feature match position for which the correlations (and hence Ψ) are close to the maximum. Inaccuracy in camera calibration means that the true hypothesis cannot reliably be identified, so the hypothesis with the highest value of Ψ is (arbitrarily) taken since the error in choosing this over the others will be relatively small. The proximity threshold test allows extra points to be reconstructed but at the cost of a loss in the accuracy of their reconstructed position.

After performing both tests, if a single verified hypothesis remains for a point, the stereo match in images 3 and 4 is recorded for use in the next cycle (eliminating the need for a search at stage A).

A disparity gradient [17] check is then applied to the *verified* stereo matches. The disparity gradient threshold is set to 0.5 and applied in both image directions, which results in essentially a general rejector of surfaces whose tangent approaches parallinity with the camera optical axes. If the image 3 to image 4 match for a point has more local point matches rejecting it under the disparity gradient test than supporting it, the point's reconstruction is rejected. This test will not affect isolated points, for which rejection and support are zero. If, in addition, it is insisted that a hypothesis has *some* support, isolated points will also be rejected. The effect of the disparity gradient test is shown in Fig. 4, for the case where some support is, and is not insisted upon.

In Fig. 3, the match between r and ii has been confirmed at the expense of the other hypothesised matches, $r - i$ and $r - iii$. The match $r - r'$ has also been confirmed, while the match $s - s'$ is found to be inconsistent, and s is matched to s'' instead.

4.2. Structure filtering

The method described in the previous section produces reliable point stereo matches which can be triangulated to estimate the position of points at consecutive time instants. For added robustness and accuracy in the estimate of this point structure it is tracked and filtered through time.

(a) (b)

Fig. 5. Images of a laboratory.

4.2.1. A stationary reference coordinate frame

The output of stereo triangulation is positioned relative to the cameras. To allow meaningful position measurement through time a fixed world coordinate frame must be used and point position tracked within this. The conversion of point position from a camera to a world coordinate frame is a function of the position and orientation of the camera in the world frame. This is defined at iteration i by a rotation of the camera, R_i, followed by a translation, t_i, relative to the new orientation. This can be updated from the *relative* movement of the camera between state $i-1$ and state i, defined by a rotation R and translation t, by the equations:

$$R_i = RR_{i-1}, \qquad t_i = R^{-1}t_{i-1} + t. \tag{6}$$

Point positions, and their uncertainty ellipsoids, can then be transferred from the camera frame (P_c, U_c) to the world frame (P_w, U_w) by

$$P_w = R_i(P_c + t_i), \qquad U_w = R_i U_c. \tag{7}$$

4.2.2. Structure filtering

A Kalman filter is used to estimate point structure position in the stationary world coordinate frame. A validation gate is used to remove points for which the reconstruction is unrepeatable. A different filter is used to track the X, Y, and Z coordinate of each point in the world coordinate frame. Each filter is a one-dimensional Kalman filter with direct measurement of the state. Position uncertainty is a function of reconstructed point position and image position uncertainty [13].

A scene reconstruction was made from a set of 50 image pairs, with large motion jumps in between them (up to 200 pixels). Two examples of left images from the sequence are shown in Fig. 5. The reconstruction is shown as a plan view in Fig. 6 – each grey blob is an output point from the program. Image position relates to world X and Z position (each square represents 1m), and the shade indicates height above the ground, lighter being higher. The observer's start position, on which the world coordinate system is defined, is at the bottom of Fig. 6, facing up. The actual layout of the room, and approximate viewing direction for the images of Fig. 5, are superimposed. Points reconstructed to a position which is at least 10 cm away from any other point are omitted. The reconstruction seems fairly reliable, no obstacles are missed, and no false structure blocks movement into unobstructed

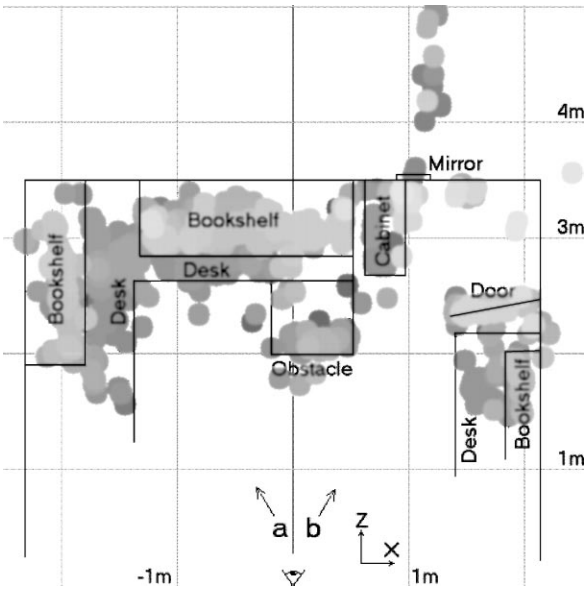

Fig. 6. Point reconstruction of a laboratory.

parts of the room. False structure is generated behind the mirror, an inherent problem while using computer vision. The images used are of size 512 × 512 pixels. As the C40 processor is too slow to process in real time, the images are processed off-line. It is expected that performance close to real time could be achieved with optimised code, smaller images, and a faster processor.

4.3. Obstacle detection

Given a reliable reconstruction and a measurement of ground plane position, either from the result of Section 3, or from a similar method applied to the point reconstruction described in this section, it is possible to identify discontinuities in the ground plane and to categorise structure as being either obstructive or harmless (lying on the ground).

5. Stair-case detection

Following the kerb detection approach [21], we start with Canny edge detection and Hough transform line-fitting on the image. For image point (u, v), the quantised Hough space is (r, θ), where θ is the angle rotated and r is the distance from the origin of the x–y coordinate system. We have

$$r = x \cos\theta + y \sin\theta,$$

where $x = u - W/2$ and $y = W/2 - v$, in which W is the dimension of the image. We accumulate evidence for straight lines from the detected edge points and then extract a small number of (r_i, θ_i)s which receive most support.

Stair-case edges are parallel to each other in 3D space. Therefore, when they are projected onto the image, these edges will intersect at a vanishing point (provided that they are not frontal parallel to the image plane). Usually a stair-case is seen from a distance, the lines will be quite parallel to each other, and the vanishing point will be far away from the image.

Based on this projective property of objects containing parallel lines, we apply our algorithm outlined below on those Hough transform fitted lines to detect and locate stair-cases in images.

5.1. The detection algorithm

The algorithm firstly picks out groups of near parallel lines, and then checks for concurrency (hence finding the vanishing point) as hypotheses for a stair-case. Then it seeks further support from the other lines for these hypotheses, to determine the best hypothesis.

Moreover, since stair-case edges are usually long and close together, the algorithm discards short edges and edges far away from the rest which are likely to be due to features other than the stair-case.

The algorithm is as follows:

1. Find k lines with maximum support from the Canny detector followed by Hough transform line fitting, discarding vertical and near vertical lines.
2. Find all groups of n lines which are within Δ degrees with each other among those k lines.
3. Check the groups for concurrency (vanishing point constraint), discarding those which do not intersect well (see below) unless the lines are horizontal whose vanishing point is at infinity, discarding also those whose intersection is within the image. If none remains, there is no hypothesis for any stair-case.
4. For each resulting group, check for more support from the other $k - n$ lines. Check how close each actual detected line is to a predicted line joining

the vanishing point found and the midpoint of the real line. Augment each group with additional supporting lines.
5. Select the group with the maximum number of supporting lines as the hypothesis for a stair-case.
6. For this group of lines, select the actual line segments which are at least of length $l_{\min}$, allowing the skipping of a couple of pixels. This caters for broken edges, but excludes short edges that could have arisen from something else. Discard line segments which are far away from the rest.
7. Output the remaining line segments as the expected stair-case hypothesis.

5.1.1. Intersection of multiple lines

We need to determine how well a number of straight lines intersect to decide whether they are considered to be concurrent or not. Theoretically, we can solve for those equations simultaneously to find the vanishing point; however, due to noise, the solution cannot be obtained for n simultaneous equations with two unknowns ($n > 2$), as those lines will not intersect perfectly with each other.

A least-squares procedure is employed to find the image point which is closest to those lines. Afterwards, the residual is computed, if it is less than a certain small threshold, then we consider these lines to be concurrent and that the image point is the vanishing point, else they are eliminated from the hypotheses.

In general, we would like to find the vanishing point $\boldsymbol{p}$ by

$$\min_{\boldsymbol{p}} C = \sum_{i=1}^{n} (\boldsymbol{l}_i^{\top} \boldsymbol{p})^2,$$

where

$$\boldsymbol{p} = [p_x, p_y, 1], \qquad \boldsymbol{l}_i = [l_{ix}, l_{iy}, l_{iz}]$$

Taking partial derivatives and setting them to zero, we have

$$\frac{\partial C}{\partial p_x} = \sum_{i=1}^{n} l_{ix}(l_{ix} p_x + l_{iy} p_y + l_{iz}) = 0,$$

$$\frac{\partial C}{\partial p_y} = \sum_{i=1}^{n} l_{iy}(l_{ix} p_x + l_{iy} p_y + l_{iz}) = 0$$

$$\Longrightarrow \begin{bmatrix} S_{xx} & S_{xy} \\ S_{xy} & S_{yy} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = - \begin{bmatrix} S_{xz} \\ S_{yz} \end{bmatrix},$$

where

$$S_{xx} = \sum_{i=1}^{n} l_{ix}^2, \qquad S_{xy} = \sum_{i=1}^{n} l_{ix} l_{iy}, \qquad S_{yy} = \sum_{i=1}^{n} l_{iy}^2,$$

$$S_{xz} = \sum_{i=1}^{n} l_{ix} l_{iz}, \qquad S_{yz} = \sum_{i=1}^{n} l_{iy} l_{iz}.$$

After solving this simultaneous system of two equations with two unknowns, we obtain (p_x^*, p_y^*), and substitute back into C to compute the residue,

$$C = \sum_{i=1}^{n} (l_{ix} p_x^* + l_{iy} p_y^* + l_{iz})^2,$$

which decides whether these lines should be regarded as concurrent or not.

In our case, the straight lines are obtained from the Hough transform and are expressed in terms of r_i and θ_i, i.e.

$$l_{ix} = \cos\theta_i, \qquad l_{iy} = \sin\theta_i, \qquad l_{iz} = -r_i.$$

5.1.2. RANSAC

Instead of finding the intersection point for groups of lines by least-squares, RANSAC can be employed. Two lines are selected from the lines, and the intersection point is found. Cases where the intersection is within the image are discarded. Support is then sought from the other lines which contain this intersection point. Repeating this procedure, the group of lines with maximum support is selected as the hypothesis for a stair-case.

Using Eq. (3), to achieve a 99% probability of a good sample in our case, assuming the percentage of contamination is 75%, the number of samples is 72.

5.1.3. Detection results

Here, we select the best 20 Hough transform lines ($k = 20$). Δ is set to 20° and $l_{\min}$ is set to 25% of the number of pixels on the actual line.

Fig. 7 shows the image for an indoor stair-case and the stair-case found. Fig. 8 shows the image for an outdoor stair-case and the stair-case found.

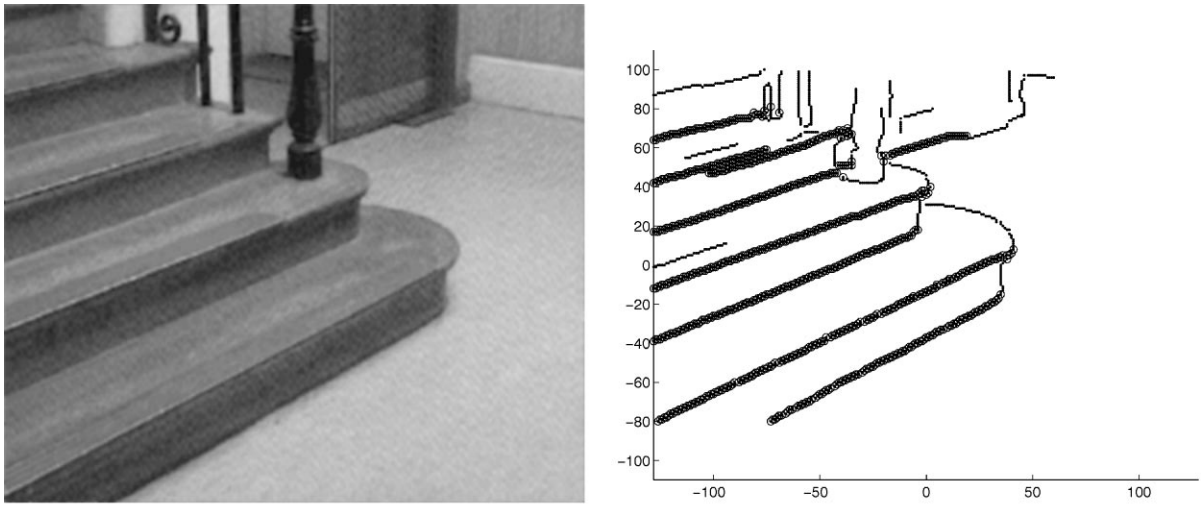

Fig. 7. An indoor stair-case, (a) the image (256 × 220 resolution); (b) the stair-case detection result with edges highlighted.

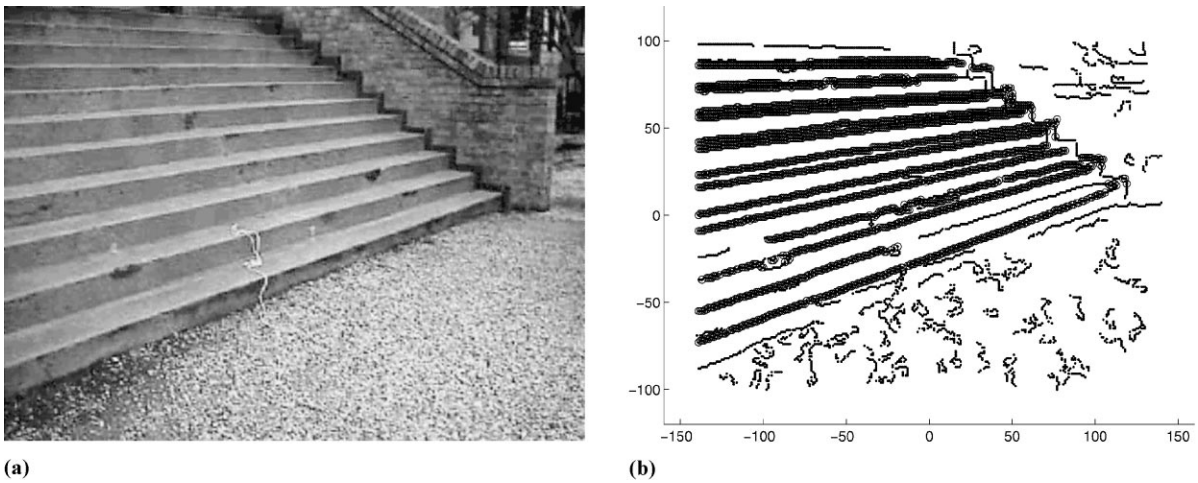

Fig. 8. An outdoor stair-case: (a) the image (320 × 240 resolution); (b) the stair-case detection result with edges highlighted.

5.2. Partition stair-case edges

So far, using the vanishing point constraint, we obtain a hypothesis for some structure containing parallel lines. We would like to add some further constraint to verify that the structure consists of two sets of equally-spaced parallel lines, the convex and the concave step edges. This is now a much stronger constraint for a regular stair-case compared to merely searching for structures with parallel lines.

The lines detected include both convex and concave edges, and so they need to be partitioned and then tested for equal spacing. The number of lines to be detected is important, using too small a number, some edges may be missing, using too big a number, many lines found will be referring to the same real edge.

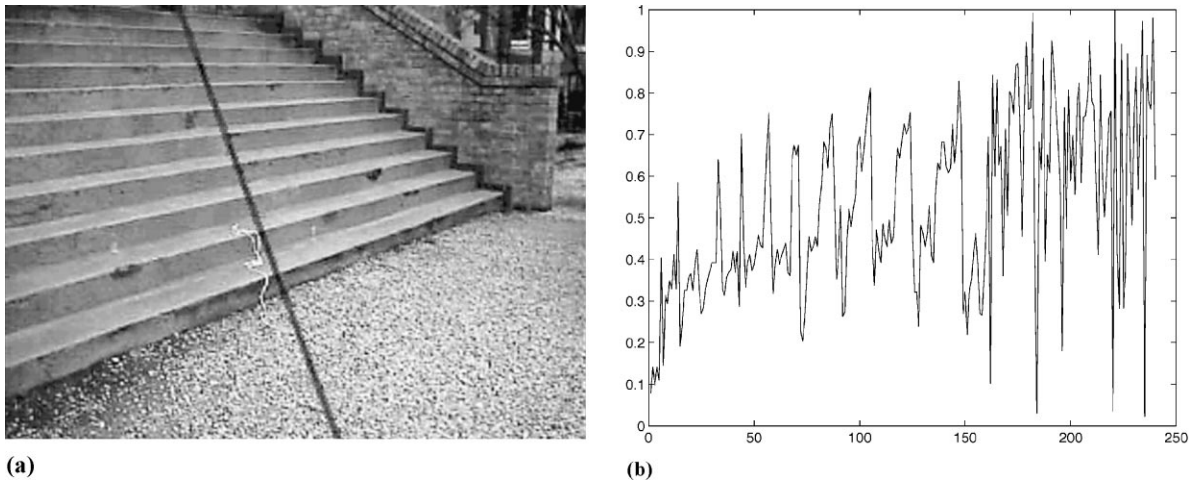

Fig. 9. Intensity variation across a stair-case. (a) A line drawn across the stair-case steps. (b) The intensity profile across the stair-case edges along this line.

5.2.1. Interleaving

Order of contact is a projective invariant we employ. For the two sets of edges (concave and convex), they interleave each other in 3D, and we can determine the order of contact when a line is drawn across them.

When this is projected onto 2D, since the order of contact with the line is the same, the two sets of lines still interleave each other in the image.

To do the partition, we look at the detected lines l_{10}, l_{20} and l_{30} when the number of lines used is 10, 20 and 30, respectively. Using l_{20} and l_{30} can help to fill in the missing edges in l_{10}. Using l_{10} and l_{20} can help to discard the extra edges in l_{30}. Combining the information from the three sets of lines, we obtain a list of lines corresponding to all the edges of the stair-case. Then, alternate lines are selected for the concave and the convex sets.

5.2.2. Cross-ratios

A cross-ratio is defined on four lines which are incident at a single point. Any set of lines incident at a common point is called a pencil. The cross-ratio of the pencil can be defined in terms of the angle between the lines. Or, for any line which cuts across the pencil, the four points of intersection define a cross-ratio on the line.

For regular stair-cases in 3D, the convex edges are equally spaced, so are the concave edges. Since cross-ratio is a projective invariant, after projection, the cross-ratio for four points on a line, each of which lies on a convex edge is 1.33, so is the cross-ratio for four points each lying on a concave edge.

After the partition, cross-ratios are then computed for each individual set. If they lie in a range around 1.33, then the hypothesised object contains two sets of equally spaced lines which are alternate to each other.

5.2.3. Line projectivity

We note that when a line is drawn across the edges of a stair-case in an image, it gives various points which correspond to points on a 3D line. Consider this line-to-line homography which is a 2×2 matrix, the d.o.f. is 3 since it is up to a scale, therefore, three points are required to find this mapping.

$$\begin{bmatrix} v' \\ 1 \end{bmatrix} = H \begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b \\ c & 1 \end{bmatrix} \begin{bmatrix} v \\ 1 \end{bmatrix} \Rightarrow v' = \frac{av + b}{cv + 1},$$

where v' is the v-coordinate along the line across the edges, obtained from the image and v is the 3D coordinate of the stair-case edge. So we select three points from the image, and we can use some consecutive integer values for v to map to each image point, e.g. 10, 11, 12.

Using these three correspondences, we can determine (a, b, c) and use them to estimate the v's for different values of v, e.g. 8, 9, 10, 11, 12, 13, 14, ... We can then check the image points obtained to seek support.

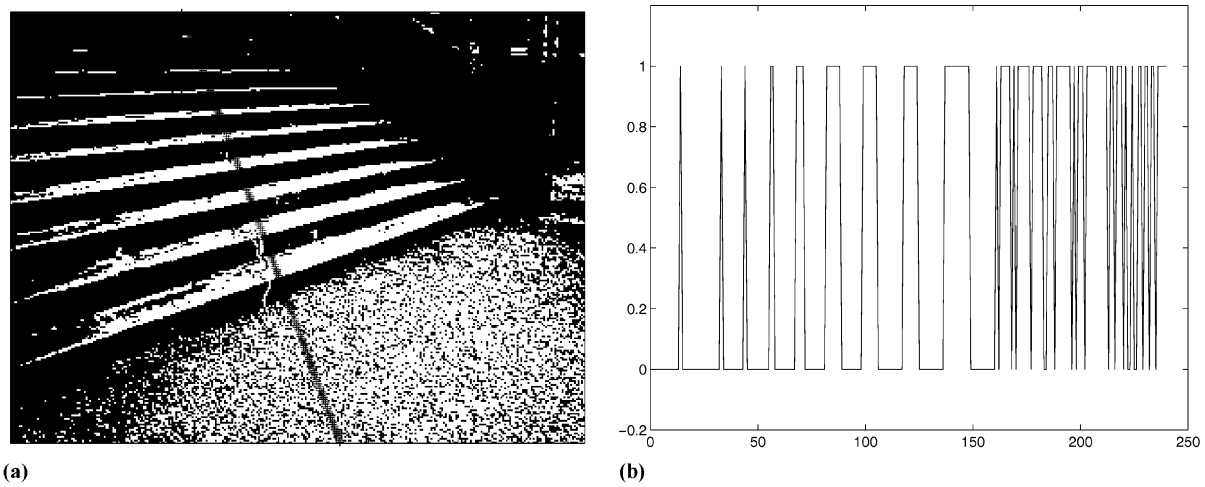

Fig. 10. Intensity variation after thresholding. (a) The line drawn across the binary image of the stair-case. (b) The intensity profile across the stair-case edges along this line.

We repeat the above either exhaustively or using RANSAC, i.e. randomly selecting three points, and select the ones with good support ($\geqslant 4$), since every selection will have at least three supporters. Each selection may potentially correspond to a group of equally spaced edges (convex or concave).

For a simple stair-case, if there are no false edges, i.e. if all the edges P are either concave or convex edges, we aim at finding partitions P_1 and P_2 such that

$$P_1 \cap P_2 = \emptyset, \qquad P_1 \cup P_2 = P.$$

Even for cases with false edges, we have instead

$$P_1 \cup P_2 \subset P.$$

Therefore, for typical stair-cases convex and concave edges are the main edges, plus perhaps a few spurious ones due to some pattern on the step for example. Among the good selections found, we need to choose two of them which do not overlap with each other and their union has a maximum size among all the other choices.

If the two selections cannot be found or the union size is far too low, that means the edges found cannot be partitioned into two sets of equally spaced lines or too many edges are missing.

Using the order of contact constraint, we can eliminate the case when there are two consecutive edges from the original set in P_1 or P_2. But we can relax the constraint a bit to allow occasional missing of edges, e.g. only eliminate sets with three consecutive edges from the original set.

5.2.4. Intensity variation

Looking at a typical stair-case image in Fig. 8(a), instead of using geometric constraints discussed above to partition the stair-case edges, we now consider the intensity variation of the stair-case as cues to do the partition.

The main idea is to detect concave and convex edges when there is a change of intensity from dark to light or from light to dark. Drawing an arbitrary line across the stair-case edges as shown in Fig. 9(a), the image intensity profile along this line will look something like Fig. 9(b).

It is clear from the profile that there exists a pattern of ups and downs of the intensity, corresponding to the tread and riser of the stair-case steps. In order to facilitate the extraction of the positions of the concave and convex edges, we use the average intensity across that line to threshold the image to obtain a binary image as shown in Fig. 10(a). The image intensity profile now will look like Fig. 10(b).

Profiling from the top towards the bottom, the concave edge occurs when the intensity changes from dark to light. On the other hand, the convex edge occurs when the intensity changes from light to dark.

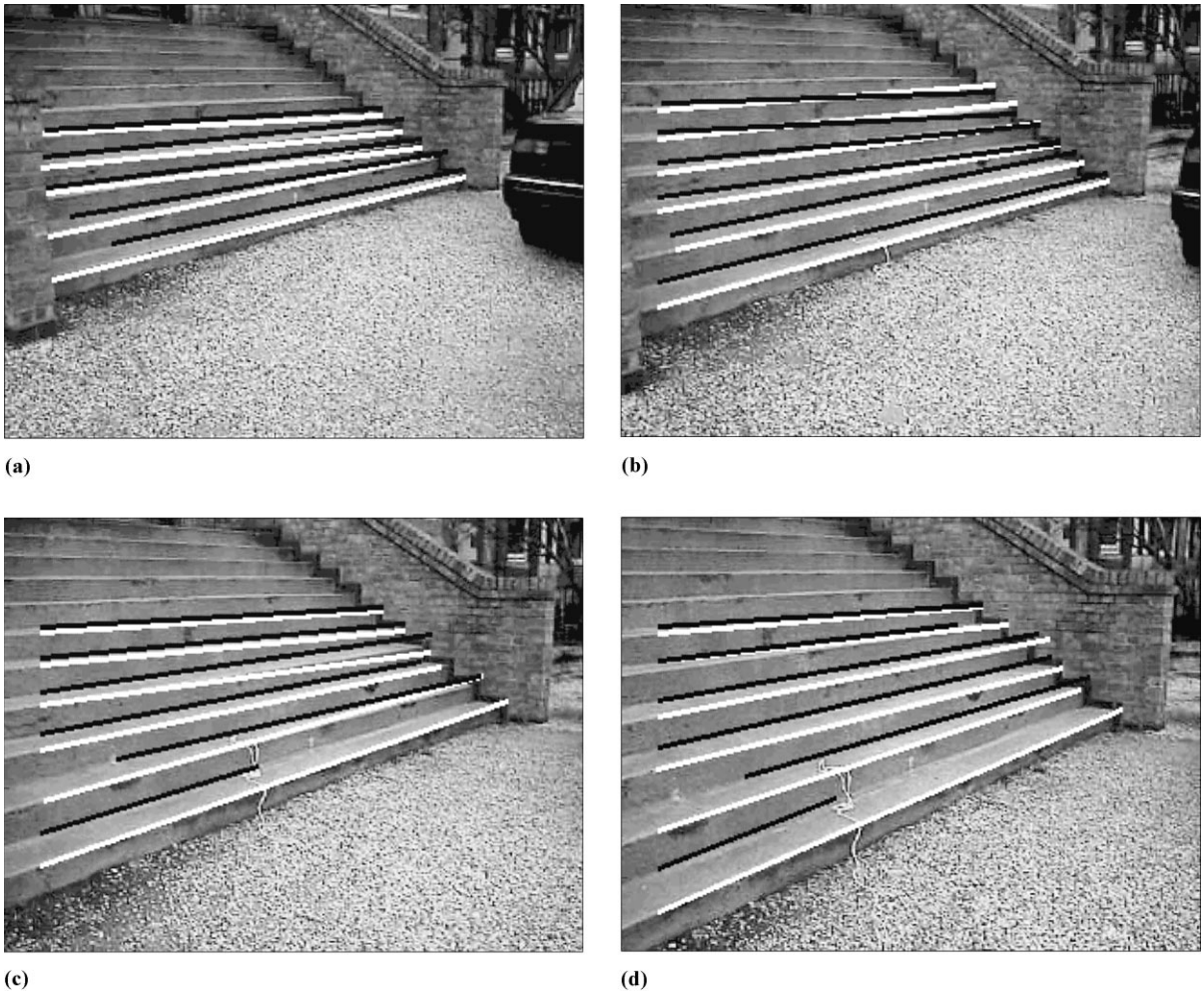

Fig. 11. Image sequence overlaid with the concave and convex edges found on the stair-case. Convex edges are marked in white while concave ones are marked in black.

Therefore, from Fig. 10(b), we can easily obtain the positions of the concave and convex edges, respectively. Geometric constraints such as cross-ratios can be applied to refine the sets as well.

5.2.5. Partition results

Verifying with cross-ratios, simply using alternate lines, is not very stable since it is very dependent on whether the number of detection lines is appropriate for a particular situation or not. Moreover, a missing edge or a spurious one will cause the following edges to be selected into the wrong group.

The line projectivity approach is better as it allows false edges, but it requires at least three consecutive convex edges and three consecutive concave edges to be found to compute the homography. By increasing the number of detection lines to avoid missing edges, we will have multiple hypotheses when there are many spurious edges. It still indicates that the structure detected contain two equally spaced sets, but based on the geometric information of the current frame alone, we cannot yet partition the edges reliably.

Here, we show in Fig. 11 a sequence of images captured while a user walks towards a stair-case. Edges

Fig. 12. Testing the user interface with the sonar sensors.

are detected using Canny edge detector and lines fitted with Hough transform. Stair-case edges are then identified with the vanishing point constraint. Afterwards, they are partitioned using the intensity variation approach described above. Concave and convex edges are overlaid on the images in Fig. 11 with convex edges marked in white and concave ones marked in black. The partition obtained using intensity variation is more stable, with edges correctly classified.

6. Testing the user interface

The user interface of the device was tested in collaboration with Irish Guide Dogs for the Blind. A number of visually impaired visitors to their training centre in Cork used a sonar-only version of the device to guide them through an 'obstacle course' similar to those used to train guide dogs and their owners. The course was about 50 m long and included a variety of obstacles that could be encountered on a city street, such as signs, barriers, fences, a bicycle, and a dustbin. Fig. 12 shows a test subject negotiating the course.

Three sonar sensors were mounted on the user's belt, one aiming directly in front of the user, and the others pointing at about 15° to each side. This configuration was found to provide adequate coverage of the 'danger area' in front of the user, and to prevent obstacles falling between the beams of neighbouring sensors. A small vibrating motor was mounted close to each sensor. Each motor was activated when the corresponding sensor detected an obstacle at a range less than a threshold selected by the user. A fourth sensor and motor were mounted at chest height.

The user interface was well received and most users found that, after a few minutes training, they were able to complete the course without collision. The use of the 'buzzers' was felt to be an intuitive way to warn the user about obstacles. The most significant training requirement was to encourage the users to rotate their bodies to scan from side to side to find a clear path, instead of stepping sideways into unknown territory.

7. Conclusions and future work

The system described in this paper demonstrates a theoretical solution to many of the problems faced by a partially sighted person trying to move around in an unknown environment.

Walk-induced motions have been successfully tracked and modelled. Reconstruction of image lines and points into entities in the world has been shown to work robustly and with sufficient accuracy. Reliable localisation of the ground plane – the only scene object which need not be avoided, and of kerbs and stair-cases – which may be negotiated with care, has also proved to be possible. The simple user interface developed has proven acceptable to users in short term evaluation tests.

Some of the vision-based parts of the system have been shown to work properly in real time, the real-time functionality of others could not be shown because of limited processing resources. More processing power would allow the real-time functionality of these parts to be demonstrated and allow the different visual parts to be linked and working together. Sensor fusion with sonar could also be realised at this point to bring considerable further benefits such as an increase in general system robustness, better localisation accuracy when both sensors detect an obstacle, and reliability under circumstances when one sensor alone misperforms. Further work on reliable staircase detection from image sequences is also planned.

The ideas presented are applicable in other areas, e.g. the guidance of a humanoid robot.

Acknowledgements

Nicholas Molton and Stephen Se are both funded by the Engineering and Physical Sciences Research Council. David Lee's research on the early stages of this system was part of ASMONC, a project funded by the TIDE initiative of the European Union.

References

[1] Y. Bar-Shalom, T.E. Fortmann, Tracking and data association, Academic Press, Boston, 1998.
[2] C.C Collins, On mobility aids for the blind, in: D.H. Warren, E.R. Strelow (Eds.), Electronic Spatial Sensing for the Blind, Martinus Nijhoff, 1985, pp. 35–64.
[3] G. Dagnelie, R.W. Massof, Toward an artificial eye, IEEE Spectrum (1996) 22–29.
[4] F. Ferrari, E. Grosso, G. Sandini, M. Magrassi, A stereo vision system for real time obstacle avoidance in unknown environment, in: Proceedings of IEEE International Workshop on Intelligent Robots and Systems (IROS '90), 1990, pp. 703–708.
[5] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography, Commun. Assoc. Comp. Mach. 24 (1981) 381–395.
[6] L. Kay, An ultrasonic sensing probe as a mobility aid for the blind, Ultrasonics 2 (1964) 53–59.
[7] P. Kovesi, Invariant measures of image features from phase information, Department of Psychology, University of Western Australia, 1996.
[8] D. Lee, The movement of sensors carried on the trunk of a walking person, Oxford University, 1996.
[9] F. Li, Visual control of AGV obstacle avoidance, D.Phil. First Year Report, Department of Engineering Science, University of Oxford, 1994.
[10] F. Li, J.M. Brady, I. Reid, H. Hu, Parallel image processing for object tracking using disparity information, in: Proceedings of the Second Asian Conference on Computer Vision (ACCV '95), Singapore, 1995, pp. 762–766.
[11] J.E.W. Mayhew, Y. Zheng, S. Cornell, The adaptive control of a four-degrees-of-freedom stereo camera head, in: H.B. Barlow, J.P. Frisby, A. Horridge, M.A. Jeeves (Eds.), Natural and Artificial Low-level Seeing Systems, The Royal Society, London, 1992, pp. 63–74.
[12] N. Molton, Egomotion recovery from stereo, D.Phil. First Year Report, Department of Engineering Science, University of Oxford, 1996.
[13] N.D. Molton, Computer vision as an aid for the visually impaired, D.Phil. Thesis, Department of Engineering Science, University of Oxford, 1999.
[14] N. Molton, S. Se, J.M. Brady, D. Lee, P. Probert, Robotic sensing for the guidance of the visually impaired, in: Proceedings of the International Conference on Field and Service Robotics (FSR '97), 1997, pp. 236–243.
[15] N. Molton, S. Se, J.M. Brady, D. Lee, P. Probert, A stereo vision-based aid for the visually impaired, Image and Vision Computing 16 (4) (1998) 251–263.
[16] S.B. Pollard, J.E.W. Mayhew, J.P. Frisby, Implementation details of the PMF stereo algorithm, in: J.E.W. Mayhew, J.P. Frisby (Eds.), 3D Model Recognition from Stereoscopic Cues, MIT Press, Cambridge, MA, 1991, pp. 33–39.
[17] S.B. Pollard, J. Porrill, J.E.W Mayhew, J.P. Frisby, Disparity gradient, Lipschitz continuity, and computing binocular correspondences, in: J.E.W. Mayhew, J.P. Frisby (Eds.) 3D Model Recognition from Stereoscopic Cues, MIT Press, Cambridge, MA, 1991, pp. 25–32.
[18] N. Pressey, Mowat sensor, Focus 3 (1977) 35–39.
[19] L. Russell, Travel path sounder, in: Proceedings of the Rotterdam Mobility Research Conference, American Foundation for the Blind, New York, 1965.
[20] S. Se, Visual aids for the blind, D.Phil. First Year Report, Department of Engineering Science, University of Oxford, 1996.
[21] S. Se, M. Brady, Vision-based detection of kerbs and steps, in: Proceedings of the Eighth British Machine Vision Conference (BMVC '97), 1997, pp. 410–419.
[22] S. Se, M. Brady, Stereo vision-based obstacle detection for partially sighted people, in: Proceedings of the Third Asian Conference on Computer Vision (ACCV '98), 1998, pp. 152–159.
[23] M. Snaith, D. Lee, P. Probert, A low-cost system using sparse vision for navigation in the urban environment, Image and Vision Computing 16 (4) (1998) 225–233.
[24] P.H.S. Torr, Motion segmentation and outlier detection, Ph.D. Thesis, Department of Engineering Science, University of Oxford, 1995.

Nicholas Molton received the BEng degree in Engineering from Brunel University in 1995. He is currently pursuing the degree of D.Phil. in the Robotics Research Group at Oxford University, and is working on the application of computer vision techniques to an aid for the visually impaired. He has published in international journals and conference proceedings. He is a student member of the IEEE, the Institute of Electrical Engineers (UK) and the British Machine Vision Association.

Stephen Se received the BEng in Computing at the Department of Computing at Imperial College, London, in June 1995. He is currently pursuing the D.Phil. degree in the Robotics Research Group at Oxford University. His research is focussed on the detection of obstacles and stairways for the partially sighted, using computer vision. He has published in international journals and conference proceedings. He is a student member of the IEEE, the Institute of Electrical Engineers (UK), the British Machine Vision Association and the British Computing Society.

Michael Brady is BP Professor of Information Engineering at the University of Oxford. Professor Brady's degrees are in Mathematics (B.Sc. and M.Sc. from Manchester University, and Ph.D. from the Australian National University). He is the author of over 250 articles in computer vision, robotics, medical image analysis, and artificial intelligence, and the author or editor of several books. Professor Brady is Editor of the journal *Artificial Intelligence*, and founding editor of the *International Journal of Robotics Research*. Professor Brady is a member of the Council of the Royal Academy of Engineering, and a former member of the UK National Technology Foresight Steering Group. He serves on the Boards of Directors of Oxford Instruments, AEA Technology, Isis Innovation. He is a founding Director of the start-up companies Guidance and Control Systems and Oxford Medical Image Analysis. Professor Brady was elected a Fellow of the Royal Academy of Engineering (UK) in 1991 and a Fellow of the Royal Society (UK) in 1997. He is a Fellow of the Institution of Electrical Engineers and a founding Fellow of the Association of Artificial Intelligence, and a fellow of the Institute of Physics. Professor Brady has been awarded honorary degrees by the universities of Essex and Manchester, and by the French CNRS.

David Lee is a Senior Lecturer in Electronic Engineering at the University of Hertfordshire, with teaching and research responsibilities in microcontrollers and embedded systems. He completed a Ph.D. in Computer Science at University College, London, in 1995, investigating the map-building and exploration strategies of a mobile robot. His thesis was selected for publication by the British Computer Society as a "Distinguished Dissertation". He then moved to the Robotics Research Group at the University of Oxford, working on electronic mobility aids for visually impaired people.

Penny Probert received the BA degree in Electrical Sciences from the University of Cambridge in 1974 and a Ph.D. in 1977. She has worked in the University of Oxford since 1988, where she is now a Reader in Engineering Science. She is a Fellow of Lady Margaret Hall, Oxford. Her research interests are in signal processing and estimation of active sensor signals (ultrasound, optical, radar), with applications mainly in navigation and robotics. She is a Fellow of the Institute of Electrical Engineers, UK, and a member of the IEEE.