

A stereo vision-based aid for the visually impaired

N. Molton*, S. Se, J.M. Brady, D. Lee, P. Probert

Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, U.K.

Received 1 November 1996; revised 3 September 1997; accepted 22 September 1997

Abstract

This paper describes a portable vision-based obstacle detection system, intended for use by blind people. The system combines an obstacle detection system designed for AGVs with recalibration of ground position and a Kalman Filter based model of the person's walking movement. The system uses stereo vision. Obstacle detection is achieved through comparison of the disparity seen with that expected from the position of the ground. Recalibration of ground position is made by plane fitting in the ground region. Motion estimation using two visual methods and the use of an inclinometer is described. The results show satisfactory success in all parts of the system. © 1998 Elsevier Science B.V.

Keywords: Obstacle Avoidance; Mobility Aids; Visual Navigation

1. Introduction

There are about two million visually impaired people living in Europe and about three million in the United States who could benefit from some form of aid to help them in their daily lives. The most obvious problem faced by the blind person is moving around in their environment without bumping into unexpected obstacles, such as a partly opened door or an object left by another person.

The possibility of restoring vision by linking cameras to the human nervous system is the subject of intensive research by both the medical and engineering professions. This approach may restore the vision of a certain proportion of blind people in the long term, but research in the area is incomplete and prototypes are not expected before 2010 [1]. There is, and there is likely to remain, the need for an external guidance device within the blind community.

1.1. Current aids for the blind

Two low-technology aids for the blind, the long cane and the guide dog, have been used by the blind for many years. A number of electronic mobility aids using sonar [2–4] have also been developed to detect obstacles, but market acceptance is rather low as the useful information obtainable from them is not significantly more than that from the long cane. They are also considered expensive and difficult to use.

On the vision side, Collins [5] described a system which converts the image point-to-point to a pattern of vibratory or electronic pulses applied to the skin. Experiments showed, however, that there is information overload for outdoor use, with background details masking primary mobility information. In addition, the response of the person to tactile stimulus soon becomes tired. The amount of user concentration required is also too great to sustain for a long time. Automatic pre-processing to provide mobility data at as high a level of abstraction as possible has been suggested [5,6], eliminating detailed clutter but retaining essential mobility information. Artificial intelligence is used to process large amounts of visual information, as a substitute to the missing functions of the eye and much of the visual pre-processing performed by the brain. In this paper we propose a visual mobility system which also provides high level information to the user, using navigation techniques demonstrated successfully in the robotics field.

1.2. Outline of system

The system we describe here provides part of an obstacle avoidance capability. It will form a major part of the mobility function of a larger project, Autonomous System for Mobility, Orientation, Navigation and Communication (ASMONC) which aims to provide a full navigation and mobility capability for blind and partially sighted people. Other sensors, such as sonar, are likely to be included to improve robustness. Sonar is reliable for detecting large and

* Corresponding author. Tel: 01865 273926; fax: 01865 273908

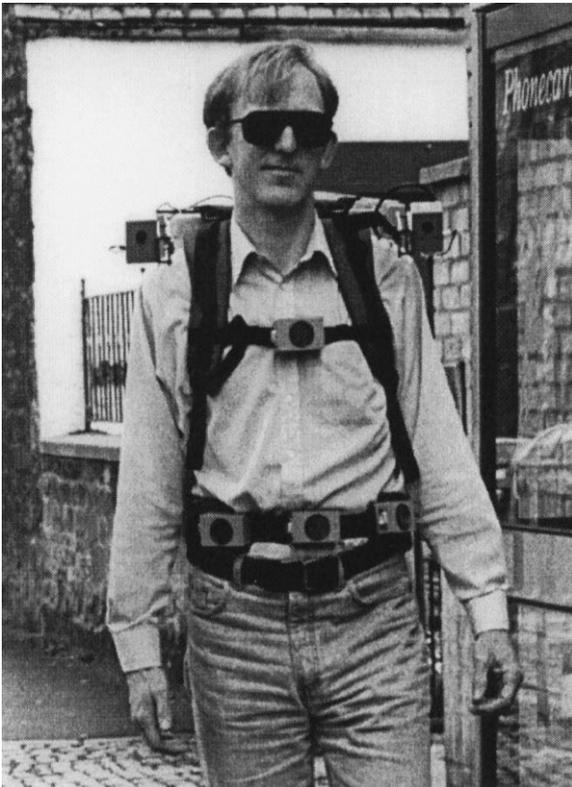


Fig. 1. The ASMONC backpack.

high obstacles, but it is not suitable for detecting small obstacles standing on the ground plane, as the angular resolution is insufficient to distinguish between the ground plane and the obstacle. Therefore a major requirement for the vision system is to detect small obstacles, and this is investigated here. We are aiming to be capable of detecting obstacles of around 10 cm in height, at a 3–5 m distance, with the vision system.

The ASMONC project uses a backpack, worn by the user, to hold the system electronics and sensors. This is shown in Fig. 1. A Loughborough Sound Images colour image processing module, based around the TI C40 processor, is used to perform all of the processing tasks required for vision. It is connected directly to two Sony NDP-40BY/E cameras. The two cameras are mounted on two rigid arms, which extend over either shoulder of the user from the backpack. Camera mounts were designed which allow minor adjustment of the camera orientation for camera alignment. The left camera is connected as the red input of the C40 module, and the right as the green. Camera calibration using Tsai's method [7] is performed. The sonar sensors are seen in the figure on the chest and belt.

The starting point of the vision system draws on the Ground Plane Obstacle Detection (GPOD) algorithm, which has been used successfully for obstacle avoidance in mobile robots by Li [8,9]. GPOD uses a pair of cameras to determine features which do not lie on the ground plane. It characterises the ground plane by a parameterisation

based on measurement of disparity, rather than projection into the external world. This improves robustness to calibration errors. It includes an initial calibration stage in which the ground plane parameters are extracted. The algorithm is described in Section 2, together with a discussion on the probable success rate for detecting different sizes of obstacle in noise.

This algorithm has been extended to include dynamic recalibration of the ground plane in successive images (Dynamic Ground Plane Recalibration, DGPR), which we describe in Section 3. Features originally known to lie in the ground are tracked, and the ground plane reparameterised at each iteration. This allows adaptation either to genuine changes in the plane's parameters or to changes in camera position. We show that the use of a Kalman Filter to track both the ground plane features and suspected obstacles provides sufficient stability for obstacle detection.

Another major requirement is the measurement and prediction of camera motion to provide the parameters which guide ground plane recalibration. In Section 4 we discuss possible approaches. The image stabilisation methods used in products such as camcorders are unacceptable, because we need to identify all six degrees of movement of the cameras, and cannot simply use an image-based readjustment. We instead discuss image-based methods for finding the six parameters of motion, and also describe briefly the use of external sensors. The measurements are fed into a Kalman Filter gait model to improve the reliability of the estimates.

2. The GPOD algorithm

2.1. Formulation of original algorithm

The GPOD algorithm proposed is a feature-based stereo matching system, which is more stable to changes in contrast and ambient lighting than region-based ones. It works in image coordinates, holds a model of the external ground plane in terms of disparity, and compares the disparity values in a new image pair with the expected ground plane disparity to detect differences (and hence obstacles). Vertical edges are detected using a Sobel detector, and stereo matching is done by PMF [10,11], including modifications to the algorithms to enhance the speed [8]. This results in fast and reliable matching. Ground images with line features on the ground but without obstacles are required for the initialisation of ground plane position.

The ground plane disparity, d , varies linearly with cyclopean image plane position Li [8].

$$d = au + bv + c$$

where (u,v) is the cyclopean image coordinate, which is midway between the left and right image coordinates. In initialisation, a least squares fit is used to estimate the ground plane parameters (a,b,c) . The measured disparity

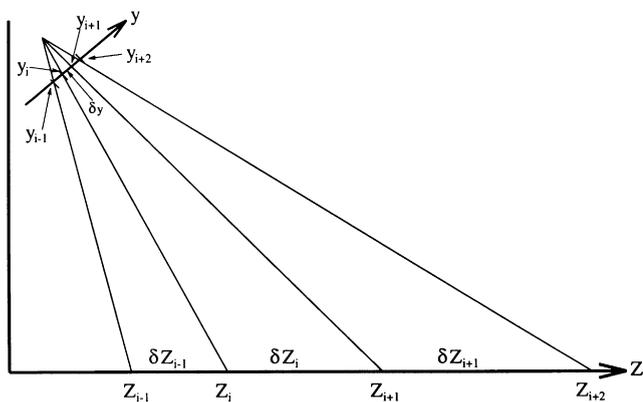


Fig. 2. This imaging geometry shows the relation between image coordinate y and ground plane distance Z with the δZ_i variation.

of an obstacle will be bigger than that which has been predicted for the ground.

For mobile robots rolling on wheels over flat ground, there is no relative change in position of the ground plane from the cameras, and hence the cyclopean ground plane disparity function is fixed. However, the cameras on the backpack are moving up and down while the blind person is walking around. The cameras can move with six degrees of freedom. Therefore, a constant initial ground plane calibration cannot be used to detect small obstacles.

We propose dynamic recalibration of the ground plane, to prevent human movement affecting obstacle detection and to obtain a better estimate of the ground plane for slopes, hills or non-flat ground.

2.2. Algorithm accuracy

In this section, we estimate the size of obstacles which might be detected. In Fig. 2, Z represents the ground plane distance (world coordinate) and y represents the vertical image coordinate. The image coordinate is a perspective projection of the camera coordinate which is a Euclidean transformation of the world coordinate. Therefore, we can express $Z = f(y)$. Since y is sampled uniformly in the image, $y_{i+1} = y_i + \delta y$ where δy is a constant, then $Z_{i+1} - Z_i = f(y_i + \delta y) - f(y_i) = \delta Z_i$.

It is intuitive that δZ_i is not constant, as the region that a pixel represents in the image depends on its position, it becomes larger as the pixel moves higher in the image as shown in Fig. 2.

The camera sampling process averages the intensity of features over δZ_i , and therefore the same feature further away will have a smaller effect on the image as a result of averaging over the larger region. Moreover, we cannot determine accurately the position of the feature as it can be anywhere in region δZ_i which increases considerably with distance. Furthermore, there are image distortions on the sides due to vignetting of the camera lens. These inaccuracies affect the ground plane disparity fitting.

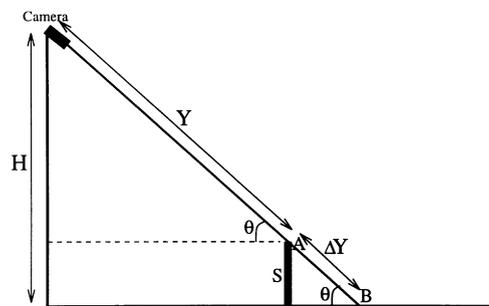


Fig. 3. The camera geometry showing the variables for various parameters.

This averaging process also implies that features in the closer region are fitted better in the initial ground plane calibration. Therefore more distant obstacles may not be detected reliably.

The disparity, d , formula is given by $d = f/Y$, where f is related to the intrinsic parameters, I is the interocular distance, and Y is the distance to the object or ground [12].

Referring to Fig. 3, the quantity used in GPOD for checking obstacles is the difference between the measured disparity at A and the predicted disparity at B, which is:

$$\Delta d = \frac{fI}{Y} - \frac{fI}{Y + \Delta Y} = \frac{fI}{Y} \left[\frac{\frac{\Delta Y}{Y}}{1 + \frac{\Delta Y}{Y}} \right]$$

From geometry,

$$\frac{\Delta Y}{Y} = \frac{S}{H - S} \Rightarrow \Delta d = \frac{fI}{Y} \left(\frac{S}{H} \right) \tag{1}$$

For our system, $I = 350$ mm, $H = 1250$ mm, and the intrinsic parameter f is approximately 300. GPOD checks whether the disparity difference of a particular pixel is bigger than some threshold. If it is, then it is marked as an obstacle. However, from the disparity data for the initial least squares fitting, a graph for the errors is obtained as shown in Fig. 4. Even without any obstacles, the disparity error at some points is up to 1.6 pixels. Therefore, we set 1.7 as our threshold to avoid false alarms. This corresponds to the disparity difference for a 10 cm high obstacle at 5 m according to Eq. (1).

2.3. Obstacle detection probability

In the previous section, we derived the disparity difference in pixels for an obstacle in purely geometric terms. However, noise makes obstacle detection a stochastic process and geometry alone does not capture the stochastic element. Therefore, we will now look at obstacle detection in probabilistic terms.

The probability of detecting an obstacle of size S is the probability of its disparity difference being larger than a certain threshold. Assuming the disparity difference is

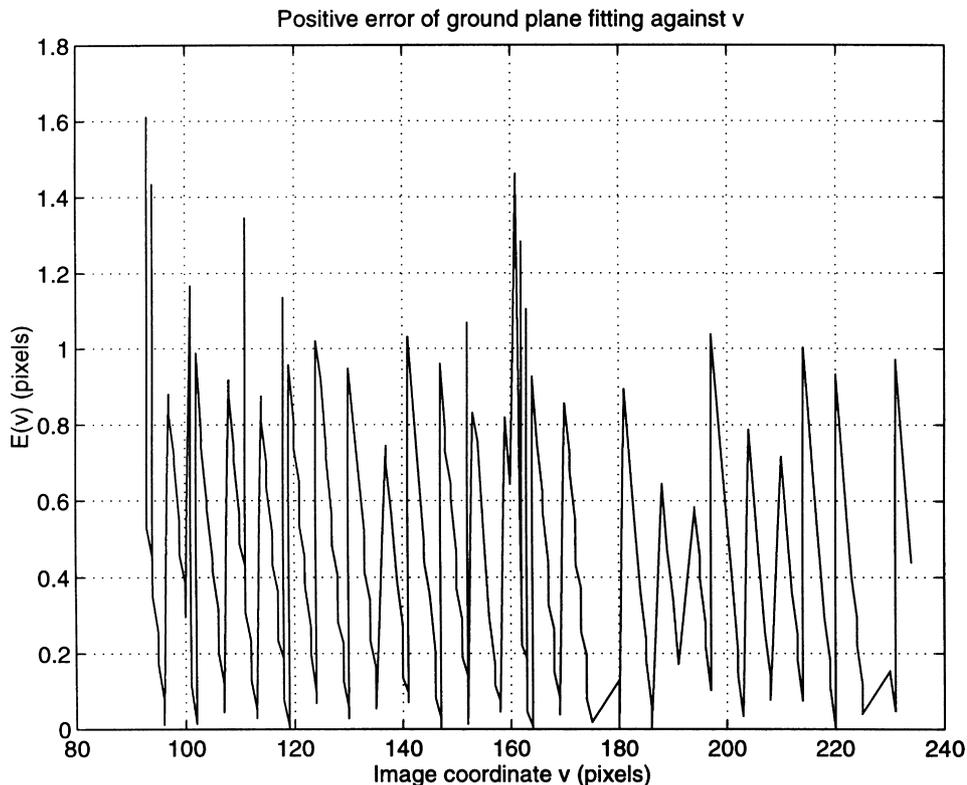


Fig. 4. Least squares error for ground plane fitting.

normally distributed, we have, using Eq. (1):

$$P(\Delta d > threshold|S) = \frac{1}{\sqrt{2\pi}\sigma_{\Delta d}} \int_{threshold}^{\infty} \frac{1}{2} \left(\frac{\Delta d - \frac{fIS}{YH}}{\sigma_{\Delta d}} \right)^2 \times e^{-\frac{1}{2} \left(\frac{\Delta d - \frac{fIS}{YH}}{\sigma_{\Delta d}} \right)^2} d(\Delta d) \quad (2)$$

where a constant $\sigma_{\Delta d} = 1$ and a constant $threshold = 1.7$ are to be used.

A vertical edge string of length n is regarded as an obstacle if at least 1 pixel of the edge string is detected as an obstacle. Therefore, if p_i is the probability obtained in Eq. (2) for the i th pixel, then the probability of obstacle detection is:

$$1 - \prod_{i=1}^n (1 - p_i)$$

We obtain Fig. 5, which shows the obstacle detection probability for different obstacle sizes at various distances. We can see that the minimum obstacle size to obtain 90% detection rate increases from 6 cm to 12 cm for distances varying from 2.5 m to 5 m.

3. Dynamic ground plane recalibration

We will describe the algorithm briefly in Section 3.1, and in more detail in Section 3.2.

3.1. The algorithm

The high-level algorithm for DGPR is as follows [13]:

Initialisation Step Use features on the ground plane without any obstacles for the initial GPOD calibration to get the parameters a , b and c .

Initiate tracks for the features on the ground plane, classified as Type I features.

Normal Step Iteratively, using step k 's ground plane parameters for obstacle detection, partition features found into 2 types:

- Type I for ground plane features;
- Type II for obstacle features.

Do track initiation, maintenance and termination for features of each type. Use the new Type I features and transformation estimate (camera movement between steps) to obtain step $k + 1$'s ground plane parameters. Use Type II feature information to send alarms to the user about obstacles in the vicinity.

The flow diagram in Fig. 6 shows the basic algorithm and the data flow between different modules. The estimated transformation input is obtained manually for the time being but will in the future be determined by the movement estimation techniques described in Section 4.

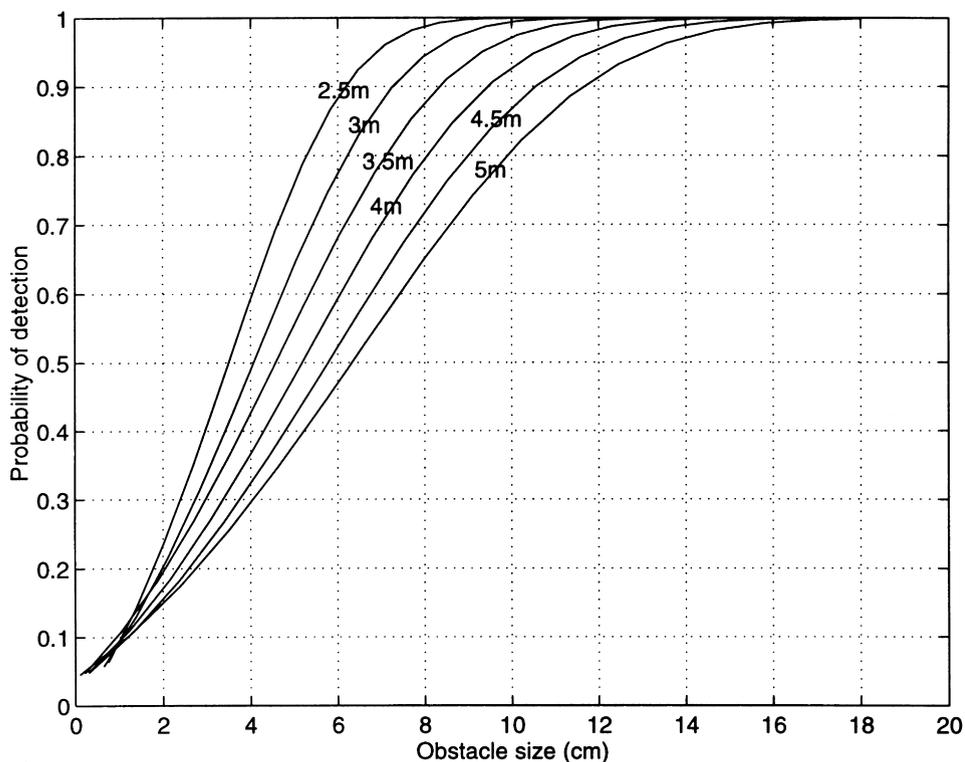


Fig. 5. Obstacle detection probability at various distances (*threshold = 1.7*).

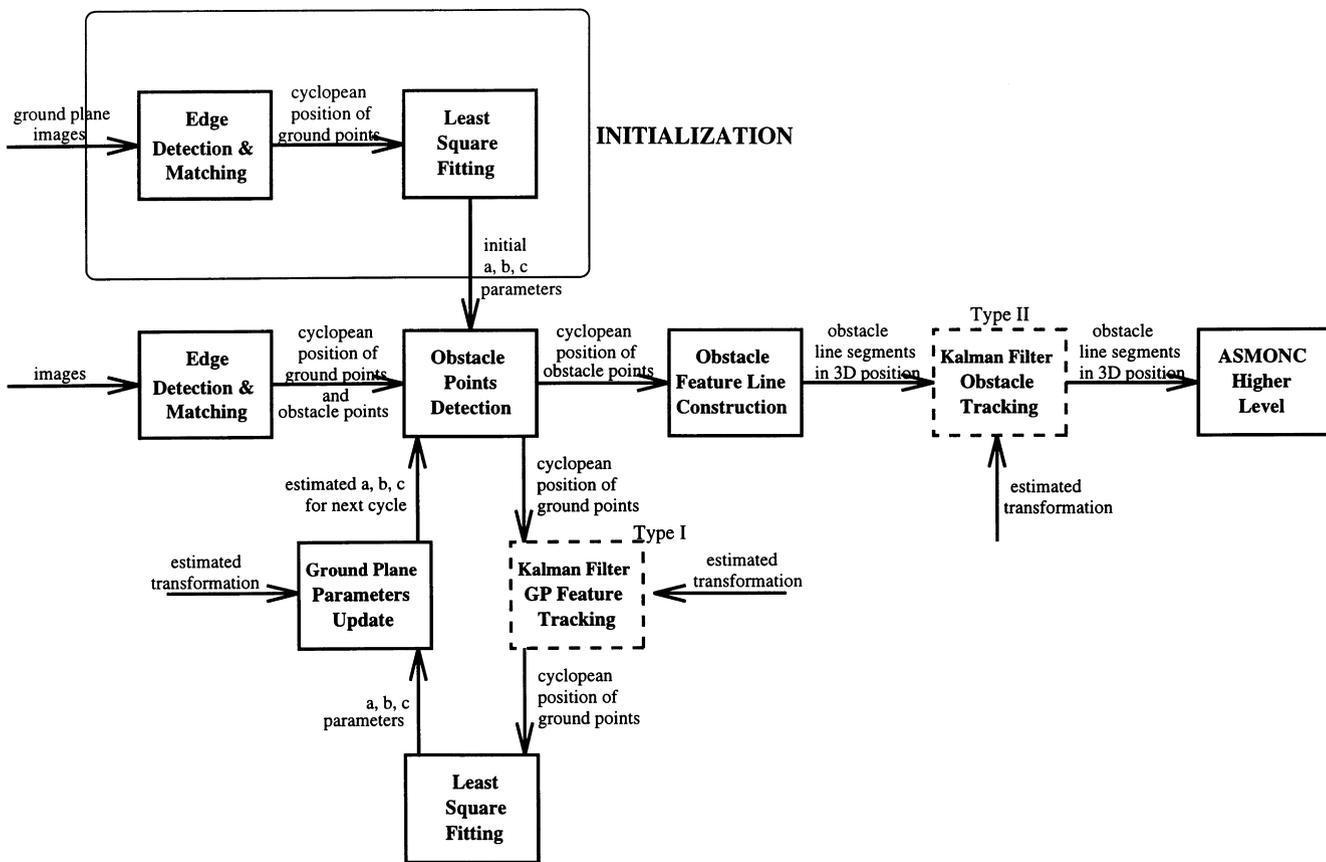


Fig. 6. Flow Diagram for the dynamic ground plane recalibration.

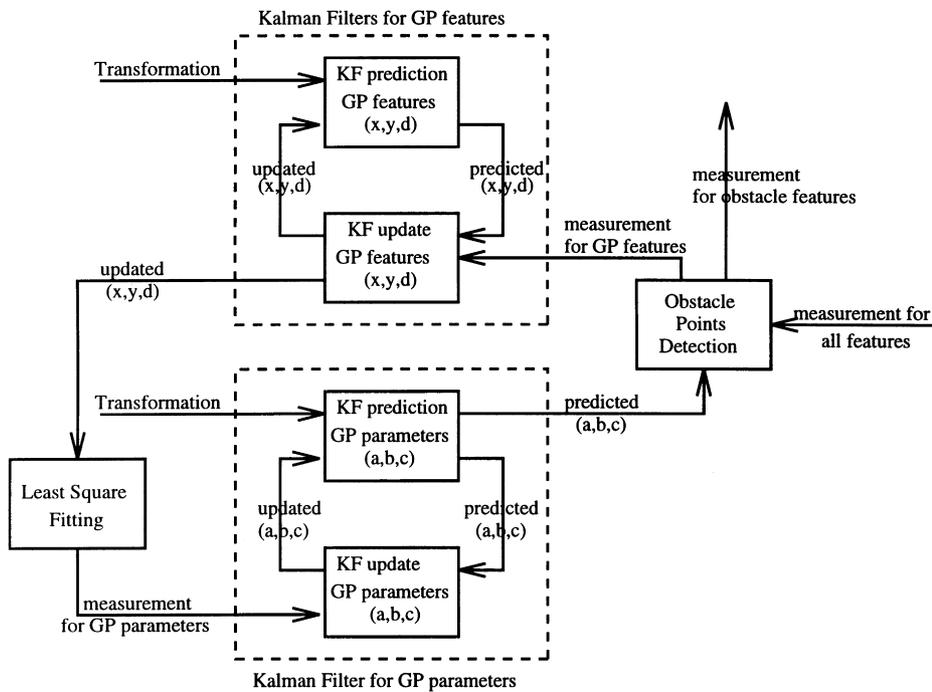


Fig. 7. Flow diagram showing the interaction between Kalman Filters for ground plane features and ground plane parameters.

3.2. Kalman Filter tracking

In a multiple-target tracking system, we have the data association problem, which addresses how to associate predictions of target positions to actual measurements. This is difficult because they may not match each other and ambiguities may arise: points may cease to exist, measurements arise from newly visible objects, spurious readings arise from noisy sensors, etc.

The Mahalanobis distance [14] is an estimate to quantify the likelihood of a measurement originating from a specific geometric feature. It is defined as

$$(\mathbf{x}_i - \boldsymbol{\mu}_i)^T (\Lambda_{\mathbf{x}_i} + \Lambda)^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_i)$$

where \mathbf{x}_i is the measurement, $\boldsymbol{\mu}_i$ is the prediction, $\Lambda_{\mathbf{x}_i}$ is the covariance of the measurement and Λ is the covariance of the prediction. This is just a generalisation of the Euclidean distance which accounts for the relative uncertainties too. It defines a validation region which provides an attention-focusing capability and reduces the complexity of data association as measurements falling outside of the region need not be considered.

The nearest-neighbour approach uses this distance metric to associate the measurements to their closest geometric features. It may occasionally perform badly as the closest measurements are not always correct [15]. Nevertheless, it is both computationally and conceptually simple, and so this approach is employed.

We propose a Kalman Filter [16] to track ground plane features as well as obstacle features. Their positions can be determined more accurately and, with suitable track initiation and termination techniques, we can deal with situations

such as new features coming into the scene, existing features leaving the scene and temporary occlusion.

We define the state of each ground plane feature to be (x,y,d) where (x,y) is the image coordinate of the feature and d is its disparity. The state of each obstacle feature is (X,Y,Z) , which are the 3-D coordinates of the tip of the obstacle.

We then use a further Kalman Filter to track the ground plane parameters in order to get a better estimate of them. Fig. 7 shows some modules of the flow diagram of Fig. 6 in greater detail and shows the interaction between the Kalman Filters. Firstly, all feature edges are detected and the Kalman Filter prediction of the ground plane parameters is used to split these edges into obstacle features and ground plane features. The ground plane features are used as the measurement for the Kalman Filter update of ground plane features. At the same time, the updated results of the ground plane features are least squares fitted, to be used as the measurement for the Kalman Filter update for the ground plane parameters.

3.3. Experimental results

We have carried out some experiments to determine the utility of DGPR. A sequence of images of a real outdoor scene was captured using the backpack. The environment is a tiled pavement with various obstacles. There was some camera motion between the images, which was measured manually. The translations were up to 20 cm, and the rotations were up to 5°, which covers the extreme case for human movement [17].

Fig. 8 shows the original image sequence where the white

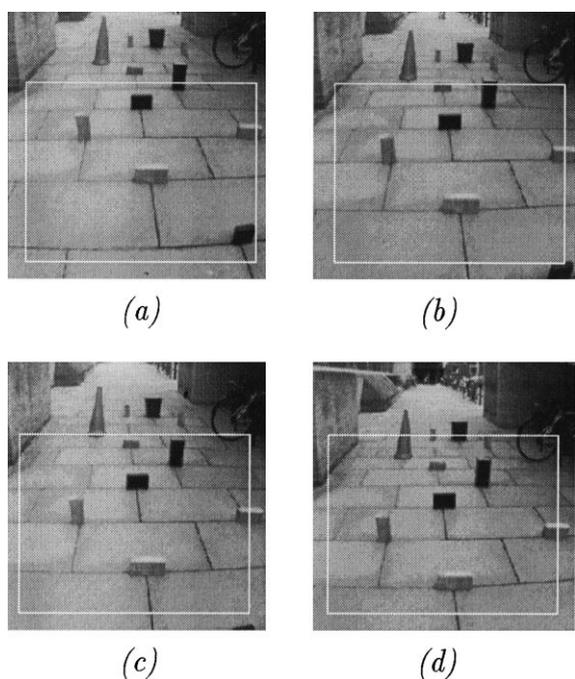


Fig. 8. The original image sequence: (a) Step 1; (b) Step 2; (c) Step 3; (d) Step 4.

rectangle indicates the window of interest. Results from GPOD and from DGPR are as shown in Figs. 9 and 10 respectively, where detected obstacles are indicated. Step 1 is at the initialisation position before any movement occurs, and after that, most obstacles are missed in GPOD but detected with DGPR. The results are tabulated in Table 1.

The positions of the obstacles detected by DGPR com-

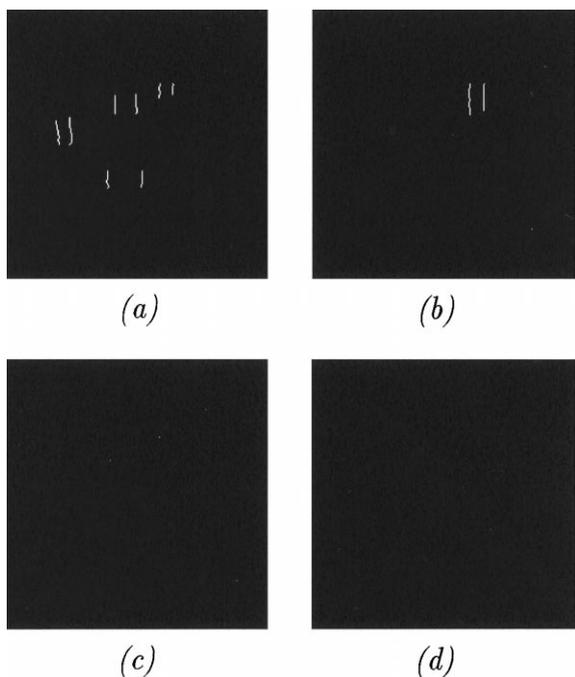


Fig. 9. Results from ground plane obstacle detection: (a) Step 1; (b) Step 2; (c) Step 3; (d) Step 4.

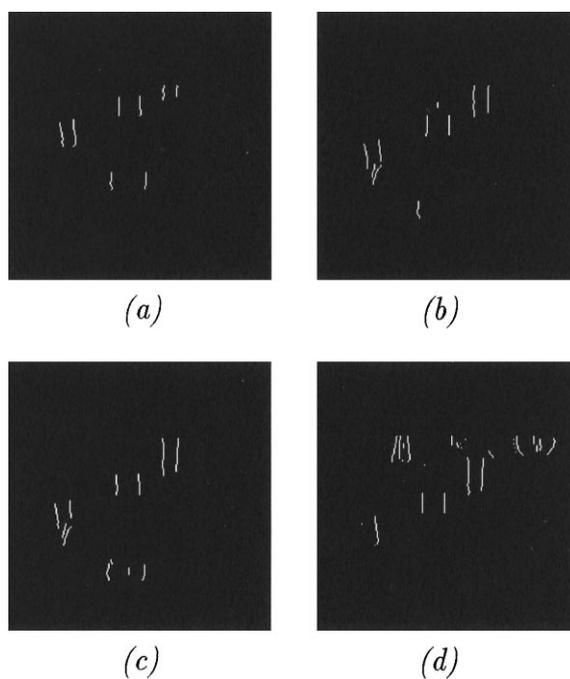


Fig. 10. Results from dynamic ground plane recalibration: (a) Step 1; (b) Step 2; (c) Step 3; (d) Step 4.

pared to their actual positions for each step are as shown in Fig. 11. It can be seen that the range estimate is not very accurate but the angular estimate is quite precise.

These results show that just using the initial ground plane parameters is not sufficient to detect obstacles in the presence of camera motion but that the DGPR approach provides some promising results. So far, the camera movement has been measured manually. The real problem still facing us is how to obtain a movement estimate to pass on to the DGPR algorithm, and so we now look at various movement estimation techniques.

4. Motion estimation

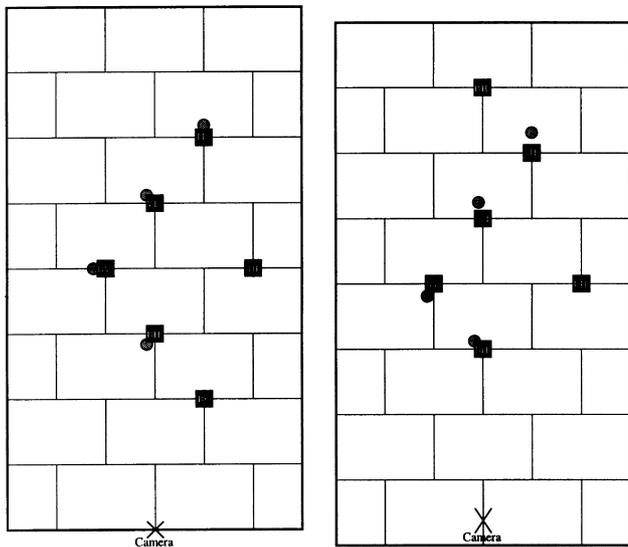
While the ground remains flat, if the motion of the cameras through space can be measured, the position of the ground in relation to the cameras can be predicted. While estimates of the pitch, roll and vertical movements are required for DGPR, predictions of all six degrees of movement are required for feature tracking. Since this is an

Table 1
Comparison between GPOD and DGPR results

Step number	1	2	3	4
Total number of obstacles in region	6	6	6	8
Number of obstacles detected by GPOD	4	1	0	0
Number of obstacles detected by DGPR	4	4	4	5

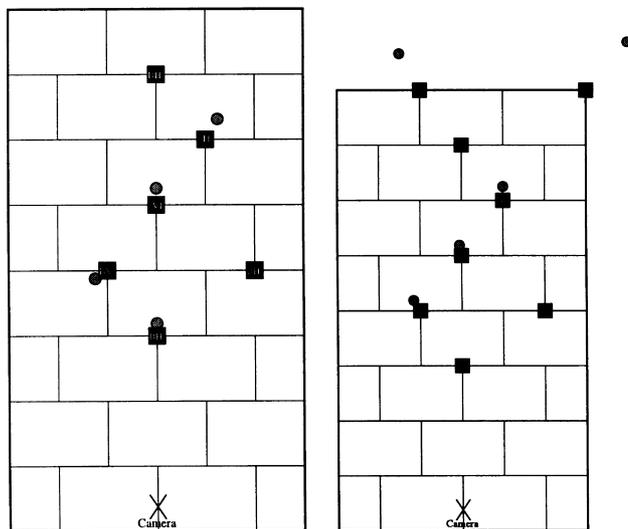
	Box - 11 cm high		Cone (part of)
	Brick (Horizontal) - 10 cm high		Bicycle Wheel (part of)
	Brick (Vertical) - 21 cm high		Wall (part of)
	Video - 15 cm high		Pavement Tile (60cm by 90cm)
	ToolBox - 33 cm high		Obstacle Detected

(a)



(b)

(c)



(d)

(e)

Fig. 11. Obstacle positions detected by dynamic recalibration against actual positions: (a) Legends; (b) Step 1; (c) Step 2; (d) Step 3; (e) Step 4.

essential requirement of the system, various methods of motion estimation were investigated and are discussed in this section.

It has been found theoretically and experimentally that the most critical motion parameter is pitch. It was estimated that a 1° error in the measured pitch is approximately equivalent to a 10 cm obstacle, 5 m away being undetectable. We

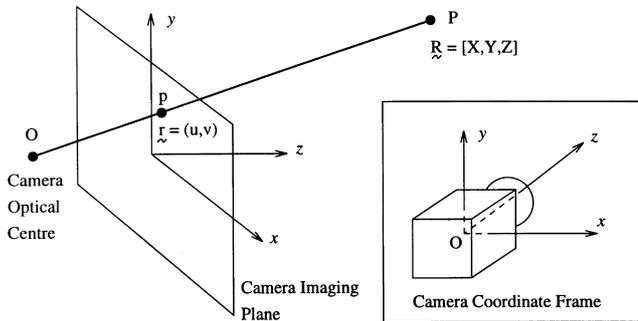


Fig. 12. The point P in the camera coordinate frame.

desire then a maximum error from the pitch estimator comparable to this, and a vertical position error smaller than the desired minimum detectable obstacle height.

4.1. Visual motion estimation

Theoretically, the motion of the backpack and cameras can be fully recovered using the images from the pair of cameras. This can be achieved through corner matching through time and between the left and right camera images, and by geometrical analysis to provide an estimate of motion.

The optic flow Eqs. (3) and (4) define the motion in the image of a point P, at $\mathbf{R} = (X, Y, Z)$, which is translated in the coordinate frame shown in Fig. 12 with velocity $\mathbf{T} = (U, V, W)$, and rotated with the angular velocity vector $\mathbf{\Omega} = (A, B, C)$. The image position of P is $\mathbf{r} = (u, v)$ and f is the focal length of the camera.

$$\dot{u} = \frac{uW - fU}{Z} + \frac{uvA}{f} - \frac{B}{f}(u^2 + f^2) + Cv \tag{3}$$

$$\dot{v} = \frac{vW - fV}{Z} + \frac{A}{f}(v^2 + f^2) - \frac{uvB}{f} - Cu \tag{4}$$

The equations show that image velocities are dependent on point movement and point 3-D position. Motion cannot be calculated without finding point depth, Z (hence position), and *vice versa*. This restricts the amount of information on motion which can be gained by looking at the position of a point in the images, meaning that many matched corners are needed to recover motion. For some applications the simplifying assumption of points at a constant depth is made. This, unfortunately is not the case for the ASMONC project, where obstacle depth varies over a large range, and the most general case must be catered for.

Corners are detected using the Wang/Brady corner detector [18], which was chosen because of its accuracy and relative speed. Detected corners are then matched by looking at the difference between the normalised intensity values within a small region around each corner in the image.

Matching is required through time in the left or right image streams and matching between the left and right

Table 2
Three point method—mean error in calculated movement

Max pitch noise	x	y	z	Roll (°)	Pitch (°)	Yaw (°)
0.1	7.0	7.5	7.9	0.05	0.14	0.13
0.5	35	37	39	0.26	0.72	0.68
1	71	74	79	0.52	1.43	1.36
2	148	153	157	1.06	2.94	2.80

images is necessary to solve the speed/scale ambiguity. In practice the motion can be estimated from a set of corners matched between the previous left and right images, and to a position in one of the current images. Knowledge of the motion and the right/left camera epipolar geometry restricts the search for the point position in the second current image to a single point (in practice a small window). Matching through time is more difficult, but point motion can be estimated to a lesser degree by using a prediction from the human walk model discussed in Section 5. When a set of seven good matches is found, the fundamental matrix [19] is calculated using a Random Sample Consensus (RANSAC) scheme [20], and is used to match the remaining points. The following two methods of egomotion estimation from the set of matched points were investigated and implemented. (For a more detailed explanation of these methods the reader is referred to Ref. [21])

4.1.1. Decomposition of the essential matrix

Since the fundamental matrix relating the left (or right) images through time is known, this can be used with the known camera properties to calculate the essential matrix [19] relating the images. The essential matrix can then be decomposed to give the camera rotation and direction of translation through time [22]. A single match between the point position in the left and right cameras at the previous instant is sufficient to calculate the absolute translation. The solution using this method becomes degenerate when a coplanar set of points is used to calculate the fundamental matrix.

4.1.2. Using partial reconstruction of three points

If the position of three points is known in both images at the last time instant (time 0) and in one of the current pair of images (time 1), the camera motion can be deduced. The previous matches allow the exact position of the three points in relation to the left camera at time 0 to be recovered, and their position at time 1 to be limited to a line in space. Since the distances and angles between the points are known at time 0, and are unchanged at time 1, the exact position of the

points at time 1 can be deduced. The transformation mapping the previous positions to the new positions can be calculated, and is the inverse of the camera motion through a static scene.

Whichever of the motion estimation methods is chosen, the calculation is made using a RANSAC scheme [20], i.e. by repeated random selection of the required number of matched points from the available set. This allows elimination of incorrectly matched points so that a least squares method can be confidently used to maximise the accuracy of the estimate.

4.2. Visual motion estimation results

An evaluation of the two methods described was first made using artificial data. A set of points in space were chosen and their positions in the image of a left and right camera were calculated. The points were then rotated (by up to 20°) and translated (a significant distance) rigidly, and new image positions found. A random amount of noise was added to the point image position before using one of the motion recovery methods on the images. The value of the noise is chosen from a uniform distribution between plus and minus a maximum pixel noise value. Table 2 shows the average error in the recovered movement parameters using the three point method for 500 tests. Table 3 shows the same information when the essential matrix method is used.

A third method was also tested, in which three points again were used but where the 3-D position at the present time is calculated from the left and right image position at this time, rather than from the known point spacings. This method, however, was found to produce higher errors than the other two.

The results show that error in the recovered motion parameters is proportional to the maximum pixel noise, hence to the localisation accuracy of the corner detector used. Slightly better results are obtained for the three point method. To reduce pitch error to below 1° the maximum pixel noise should be at around 0.5 pixels.

Table 3
Essential matrix method—mean error in calculated movement

Max pixel noise	x	y	z	Roll (°)	Pitch (°)	Yaw (°)
0.1	9.95	9.03	12.14	0.11	0.17	0.18
0.5	45	43	60	0.46	0.82	0.84
1	84	81	118	0.92	1.48	1.60
2	163	149	241	1.78	2.42	2.59

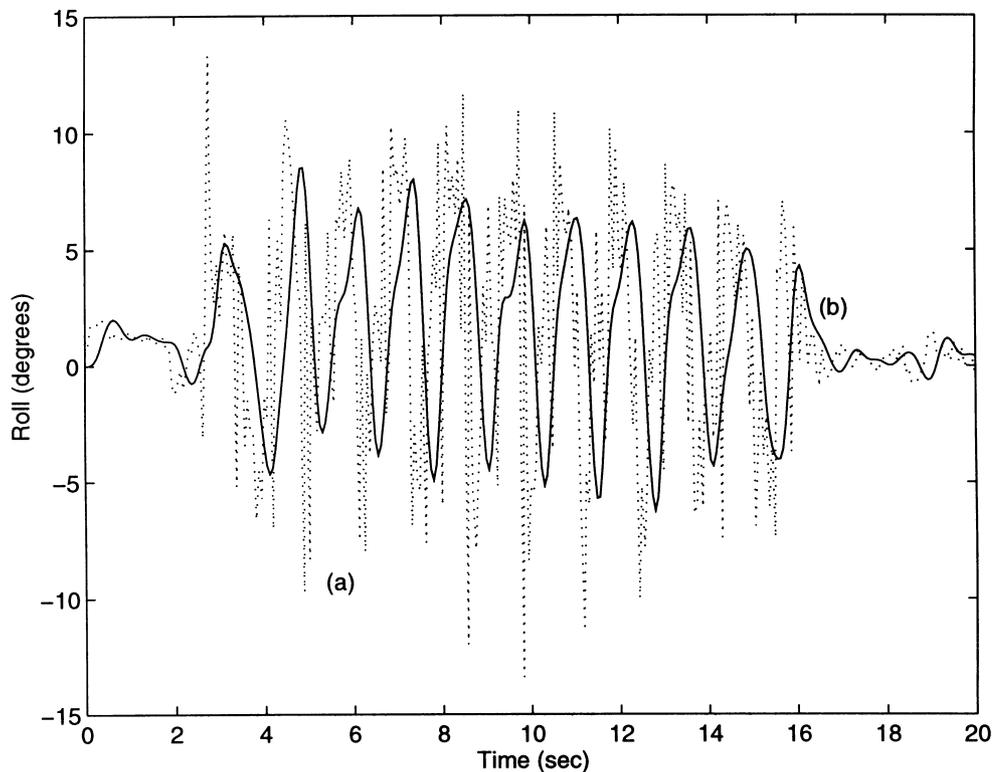


Fig. 13. Roll during walking: (a) sensor output; (b) filtered.

The two methods of motion estimation described were also tested on a number of stereo image sequences from outdoor and indoor scenes. The error in calculated movement between frames here was typically similar to the values for a maximum pixel noise value of one. The Wang/Brady corner detector in the form used here could not be expected to locate corners consistently to sub-pixel accuracy.

The accuracy of the method is satisfactory, however it is computationally expensive and so cannot easily be implemented in real time. The measurement is also incremental, rather than absolute, meaning that when used with dynamic recalibration the motion estimate error is cumulative between recalibration points, and so could periodically become very large. The effect of this can be reduced by using a Kalman Filter, as described in Section 5, and making assumptions about the underlying constancy of the motion.

These two drawbacks, with our current setup, initially limited the use of vision as input to a motion tracking Kalman filter for obstacle detection. However, a different form of motion sensor was available for experiments in the form of a digital compass/inclinometer module, which measures roll, pitch and yaw. This sensor has the significant disadvantage of introducing a phase lag into the signals measured but can work in real time. The sensor was considered suitable for an initial study of motion using a Kalman Filter, and this is described in Section 5. The ideal solution is likely to be the use of both a visual, and a

non-visual sensor as input to the Kalman Filter, which should make available the benefits of both.

5. Human walk model

A Kalman Filter based model was devised to track the walk motions, given available sensor measurements (at this stage from the digital compass). The purpose of the filter is to improve the accuracy of the movement estimates and give an estimate of the expected movement in the near future.

The walk dynamics model is based on the results of Lee [17], who analysed the movement of someone walking, carrying a load, in a gait lab. Fourier analysis of the results obtained shows that the six components of movement of a plane centred around the person's shoulders are each predominantly at a single frequency. Furthermore, the roll and yaw frequency are identical, and are half the pitch frequency. Similar results were reported by Inman et al. [23].

During the experiment, the sensor module was fitted rigidly to a backpack, and measurements taken while a person wearing the backpack walked in a straight line. The resulting signals were noisy but after filtering with a digital Butterworth filter seemed to be consistently repetitive and close to sinusoidal. Fig. 13 shows a typical roll signal from the sensor. The pitch signal appears similar while the yaw signal is of constant period but more variable amplitude. The signals clearly describe the movement pattern of the person and demonstrate potential for tracking.

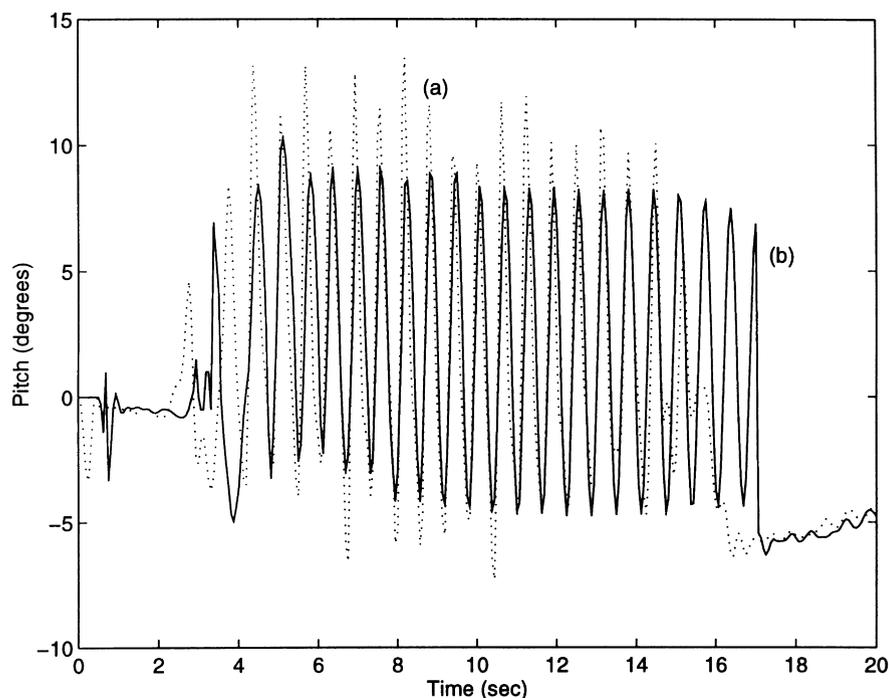


Fig. 14. During walking: (a) measured; (b) predicted half of a second earlier.

Each component of movement is modelled by a single sine wave with variable parameters of amplitude, phase, and a fixed offset from zero, which are assumed to be constant over short periods of time. A common base frequency is used for the three motion components. The effect of this is to improve the filter convergence time and robustness to measurement error. Using this model an extended iterated Kalman Filter was implemented to track the parameters, using the inclinometer data alone. The Kalman Filter prediction of future movement is shown for the pitch component in Fig. 14. A simple change detector is used to identify when the walk model movements exist. This is used to detect walk starts, stops, or non-walking movements (such as sitting down). When no walking is detected the state estimates are frozen and the current position becomes the prediction.

The Kalman Filter was then modified to allow more flexibility in the states. The resulting filter is able to deal with typical real-life motions which show variation in speed, corner turning, and walking uphill, despite the fact that at this stage no attempt is made to detect these situations, as is done when the walker stops. The prediction of yaw under such a scenario is shown in Fig. 15. The flexibility of the filter means that its parameters do not need to be 'tuned' for specific users or conditions. Because only three measurements are used the filter works at the sampling rate of the inclinometer (16 Hz).

Further analysis is underway to determine the relative phase of the components and perhaps find the relationship between their amplitudes, which would simplify the Kalman Filter implementation considerably. Measurements of translation might also be included from another sensor or the

visual motion recovery method, as this is also required for DGPR.

It is also intended that the measurement inputs to the Kalman Filter be compared to the movements of the person as measured in a gait lab, to check for differences particularly in the measurement phase and amplitude. Compensation for the phase lag introduced by the filter is also required, but is fairly straightforward as the signals are largely at one frequency, and the phase response of the filter is known.

The state dynamics model described above might be dynamically adjusted to make use of higher level comments from ASMONC to the blind user. For example, if the system stops the user, warning might be given to the filter program which will initiate a constant deceleration kinetic model. The model might also be augmented to include a measure of user turning rate, in which case guidance information from ASMONC might trigger a constant turning rate model. Clearly, filter performance may be improved if a change in the type of motion (particularly cornering) is detected (either visually or by observation of filter parameters), and a movement model chosen accordingly.

The visual motion recovery method is not sufficiently fast at present to maintain the movement model alone in real time, but its measurements can be combined with those from the inclinometer in the filter to improve measurement accuracy. It would also be possible to further augment the measurement vector at a later date to add information from dynamic recalibration or the other ASMONC sensors.

The results show that a scheme based on a combination of vision and non-visual devices would be sufficiently accurate

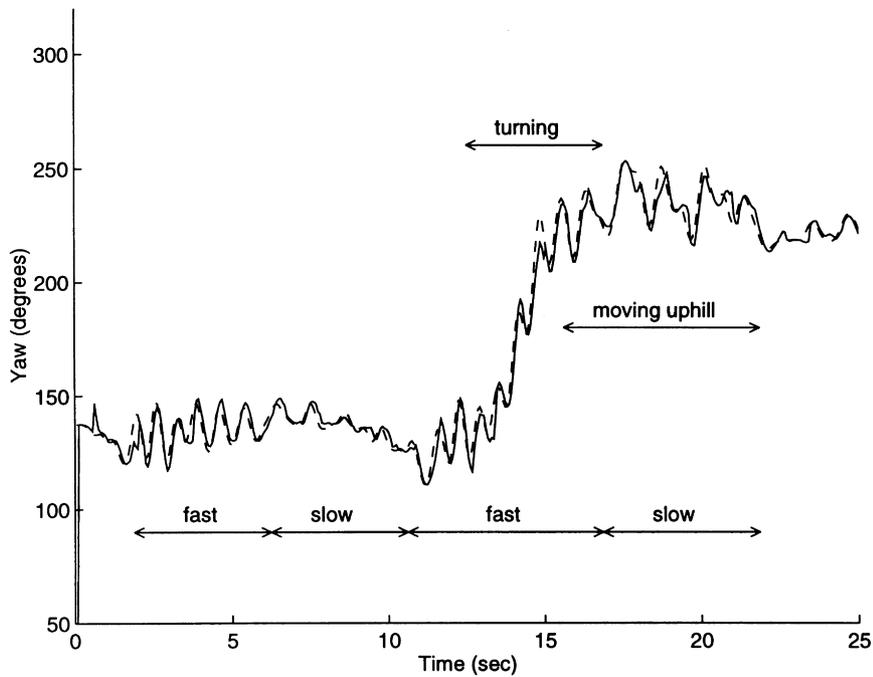


Fig. 15. During complex walk: solid line shows measurement, dotted shows prediction made half of a second earlier.

to fulfil the requirements of the DGPR obstacle detection process.

6. Performance

The DGPR is not running in real-time yet with the current C40 processor, which is a few years old and very slow. We compare the processing speed on the 128×128 image sequence shown earlier with various processors, and the results are tabulated in Table 4. The image capturing time with our current hardware is around 200 ms per pair. It can be seen that a practical system of 2 Hz can be easily achieved with a Pentium 166 processor.

7. Conclusion

In this paper we have described the GPOD algorithm, together with some analysis of its accuracy in the case where the ground plane position is more or less the same relative to the cameras. However, for the ASMONC case,

we require dynamic recalibration of the ground plane due to the walking motion of the person. We present the algorithm and some experimental results carried out in an outdoor scene in Section 3, showing that the latter approach is quite promising. The estimate of camera movement required for dynamic recalibration has so far been obtained manually, but Sections 4 and 5 describe the procurement of this information via visual tracking and gait analysis, as well as by compass/inclinometer. The results to date show that a vision based obstacle detection system for blind people is feasible.

The work presented here is part of an ongoing research project, and we are currently looking into various issues. We are investigating ways to speed up the algorithm, such as parallelisation and feeding information forward between frames to constrain the window size. As always, there is a trade-off here between performance and cost/power consumption, which must be carefully investigated. For the determination of camera movement, we are currently pursuing a more accurate corner localisation method, and investigating the use of assumptions about the world to simplify motion calculation.

The work described here is applicable to any situation where there is movement of a pair of cameras over ground. If there is any known pattern to the movement, it could be modelled and used to improve accuracy and reliability of the recovered egomotion. Examples to which it might be particularly suited are when stereo cameras are attached to a human being, a legged robot, or a wheeled robotic vehicle on bumpy ground. In these cases visual motion recovery may be appropriate, and hence DGPR can be used for obstacle detection and navigation. Where the movement is

Table 4
Comparison of Processing Speed for DGPR on 128×128 Image Pair

Processor	Processing time per image pair (s)	Speed (Hz)
C40	7	$\frac{1}{7}$
Ultra-Sparc	1	1
Pentium 166	0.5	2
SGI	0.25	4

periodic or has some pattern to it, a model of the known movement will certainly be beneficial.

Acknowledgements

The research of the first two named authors is funded by the EPSRC. The ASMONC project is funded by the European Union under the TIDE initiative.

References

- [1] G. Dagnelie, R.W. Massof, Toward an artificial eye, *IEEE Spectrum* 33(5) (May) (1996) 22–29.
- [2] L. Russell, Travel path sounder, in: *Proceedings of the Rotterdam Mobility Research Conference*, American Foundation for the Blind, New York, 1965.
- [3] N. Pressey, Mowat sensor, *Focus* 3 (1977) 35–39.
- [4] L. Kay, An ultrasonic sensing probe as a mobility aid for the blind, *Ultrasonics* 2 (1964) 53–59.
- [5] 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 Publishers, Dordrecht, 1985, pp. 35–64.
- [6] M.F. Deering, Computer vision requirements in: blind mobility aids, in: D.H. Warren, E.R. Strelow (Eds.), *Electronic Spatial Sensing for the Blind*, Martinus Nijhoff Publishers, Dordrecht, 1985, pp. 65–82.
- [7] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off the shelf TV cameras and lenses, *IEEE Journal of Robotics and Automation* 3 (4) (1987) 323–342.
- [8] F. Li, Visual control of AGV obstacle avoidance, DPhil First Year Report, Department of Engineering Science, University of Oxford, 1994.
- [9] F. Li, J.M. Brady, I. Reid, H. Hu, Parallel image processing for object tracking using disparity information, in: E.K. Teoh (editor), *Second Asian Conference on Computer Vision ACCV '95*, Singapore, December 1995, pp. 762–766.
- [10] 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, Cambridge, Massachusetts, 1991, pp. 33–39.
- [11] 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, Cambridge, Massachusetts, 1991, pp. 25–32.
- [12] B.K.P. Horn, *Robot Vision*, MIT, Cambridge, Massachusetts, 1986.
- [13] S. Se, Visual aids for the blind, DPhil First Year Report, Department of Engineering Science, University of Oxford, 1996.
- [14] Z. Zhang, O. Faugeras, *3D Dynamic Scene Analysis*, Springer-Verlag, Berlin, 1992.
- [15] I.J. Cox, A review of statistical data association techniques for motion correspondence, *International Journal of Computer Vision* 10 (1) (1993) 53–66.
- [16] Y. Bar-Shalom, T.E. Fortmann, *Tracking and Data Association*, Academic Press, Boston, London, 1988.
- [17] D. Lee, *The movement of sensors carried on the trunk of a walking person*, Oxford University, January 1996.
- [18] H. Wang, J.M. Brady, Real-time corner detection algorithm for motion estimation, *Image and Vision Computing* 13 (9) (1995) 695–703.
- [19] O.D. Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig? in: G. Sandini (Eds.), *Proceedings of 2nd European Conference on Computer Vision*, Springer-Verlag, Berlin, pp. 563–578, 1992.
- [20] 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.
- [21] N. Molton, *Egomotion recovery from stereo*, DPhil First Year Report, Department of Engineering Science, University of Oxford, 1996.
- [22] O. Faugeras, *3 Dimensional Computer Vision—A Geometric Viewpoint*, MIT Press, Cambridge, Massachusetts, 1993.
- [23] V.T. Inman, H.J. Ralson, F. Todd, *Human Walking*, 1st edn, Williams and Wilkins, Baltimore, 1981.