Mobile Robot Localisation using Active Vision

Proc 5th European Conference on Computer Vision, Freiburg, Germany,

1998pp809–825 (Springer LNCS Vol 1407, Volume II)

Andrew J Davison and David W Murray

Department of Engineering Science, University of Oxford,
Parks Road, Oxford OX1 3PJ, UK
[ajd|dwm]@robots.ox.ac.uk

Abstract. Active cameras provide a mobile robot with the capability to fixate and track features over a wide field of view. However, their use emphasises serial attention focussing on a succession of scene features, raising the question of how this should be best achieved to provide localisation information. This paper describes a fully automatic system, able to detect, store and track suitable landmark features during goal-directed navigation. The robot chooses which of the available set of landmarks to track at a certain time to best improve its position knowledge, and decides when it is time to search for new features. Localisation performance improves on that achieved using odometry alone and shows significant advantages over passive structure-from-motion techniques. Rigorous consideration is given to the propagation of uncertainty in the estimation of the positions of the robot and scene features as the robot moves, fixates and shifts fixation. The paper shows how the estimates of these quantities are inherently coupled in any map-building system, and how features can reliably be re-found after periods of neglect, mitigating the "motion drift" problem often encountered in structure-from-motion algorithms.

1 Introduction

Active cameras potentially provide a navigating vehicle with the ability to fixate and track features over extended periods of time, and wide fields of view. While it is relatively straightforward to apply fixating vision to tactical, short-term navigation tasks such as servoing around obstacles where the fixation point does not change [6], the problem of using serial fixation on a succession of features to provide global localisation information for strategic navigation is more involved.

In this paper, we demonstrate a system which is able to detect, store and track suitable landmark features during goal-directed navigation. The features used are arbitrary points of high contrast which are abundant in any typical environment. The robot chooses which of the available set of landmarks to track at a certain time to best improve its position knowledge, and decides when to change fixation point and when to search for new features with the aim of maintaining a useful 3D map. When suitable landmarks have been detected, the robot is able to refer to them at any time to calculate its position, or to track one of them during a movement to provide continuous information. The method has significant advantages over the passive structure-from-motion techniques used in a number of navigation systems [1–4]. Parts of our approach could

be considered as an transfer to the visual domain of methodology used in directable sonar work [5].

Navigating in unknown surroundings inherently couples the processes of building a map of the area and calculating the robot's location relative to that map. With this in mind, a single filter is used to maintain estimates of all the quantities of interest and the covariances between them. This approach differs from many previous approaches to robot navigation and structure-from-motion (e.g. [3, 4]) where separate filters are used for the robot position and that of each of the features. In the work in this paper, the goal is to provide the robot with a sparse map of features which can be used for localisation over extended periods during which the same areas may be traversed many times and the same features will be observed. Only full uncertainty propagation will lead to the localisation performance we should expect in these circumstances, as will be shown in the experiments described later. Our approach allows features to be re-found and re-registered reliably after periods of neglect. This alleviates the problem of "motion drift" often encountered in structure-from-motion, where, for instance, the start and end points of a closed path are not recognised as such [3].

When deciding upon the exact form of the filter, the issue of coordinate frames and their significance was considered in detail. Consider navigation around a restricted area such as a room: certainly there is no need to know about position or orientation relative the world as a whole, but just about the relative location of certain features of the room (walls, doors, etc.). This initially suggested a completely robot-centred approach to navigation, attractive as is minimises the problem of representation. However, such an approach cannot explicitly answer questions such as "how far has the robot moved between points A and B on its trajectory?", important in goal-directed performance. Our approach explicitly estimates the vehicle and feature positions relative to a world frame while maintaining covariances between all the estimates. This provides a way to answer these questions while retaining all the functionality of the robot-centred method, since it is possible at any stage to re-zero the coordinate frame to the robot, as we will show in Section 3.6. The extra information held in the explicit robot state and its covariances with the feature estimates codes the registration information between an arbitrary world coordinate frame or map and the structure of the feature map the robot has built itself.

2 Vehicle and Head Geometry

2.1 Vehicle Geometry and Kinematics

The robot used in this work (Figure 1) has three wheels: two run freely on a fixed transverse axis at the front, while the third, located centrally at the rear, is both steerable and driven. A high performance four-axis active stereo head [7] is mounted with its vertical pan axis directly above the point halfway between the front wheels. The fiduciary "head centre" is defined to lie where the pan axis intersects the horizontal plane containing the elevation axis. This point, fixed relative to the vehicle regardless of head movements, is used to define the vehicle's location relative to a world coordinate frame. The robot vehicle is assumed to move at all times on the horizontal xz ground-plane. Its position and orientation are specified by the coordinates (z, x, ϕ). ϕ is the robot's orientation relative to the world z-axis. The robot-centred frame $C0$ is defined to have its origin on the ground plane directly under the head centre, with its z-axis pointing to the front of

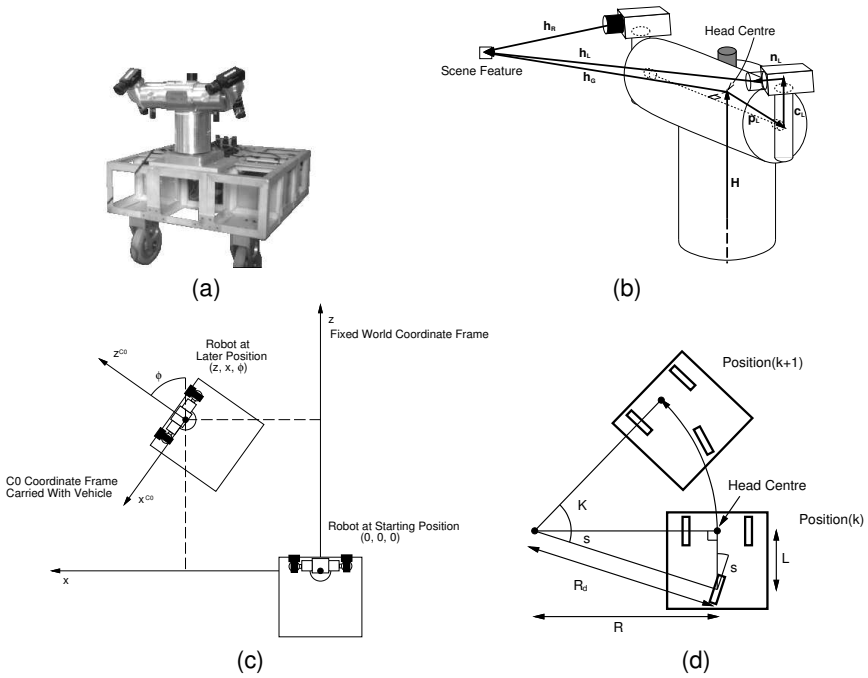

Fig. 1. Head and vehicle geometry. In (c) the vehicle's location in the world coordinate frame is specified with the coordinates (z, x, ϕ). The $C0$ coordinate frame is carried with the vehicle.

the robot, x-axis to the left, and y-axis upwards. In normal operation, at the start of its motion the vehicle's location in the world frame is defined as $z = 0$, $x = 0$, $\phi = 0$, and the world and vehicle frames coincide.

The control inputs determining the vehicle motion are the steering angle s and velocity v of the rear driving wheel. With no slippage, setting a steering angle of s at the rear wheel means that points on the vehicle will travel in circular paths centred on the centre of rotation at the intersection of the wheel axes, as shown in Figure 1(d). In particular, the head centre and rear driving wheel move in paths of radius $R = L/\tan(s)$ and $R_d = L/\sin(s)$ respectively. During a period Δt in which both v and s are held constant, the angle in radians through which the vehicle moves along its circular trajectory is $K = v\Delta t/R_d$, after which the new location of the head centre becomes

$$
\begin{aligned}
z(t + \Delta t) &= z(t) + R(t)\,(\cos\phi(t)\sin K + \sin\phi(t)(\cos K - 1)) \\
x(t + \Delta t) &= x(t) + R(t)\,(\sin\phi(t)\sin K - \cos\phi(t)(\cos K - 1)) \\
\phi(t + \Delta t) &= \phi(t) + K\ .
\end{aligned}
$$

These are exact expressions and do not require Δt to be small. (Note that for straight line motion ($s = 0$, R and R_d infinite) a limiting form is used.) The robot will of course not move precisely in the way predicted by the equations due to factors including slipping of the wheels and the non-zero time taken to respond to commands. This

uncertainty is modelled as a Gaussian variation in v and s from the demanded values. The steering angle s, was given a constant standard deviation σ_s of around 4°, and the velocity input v a standard deviation which was proportional to the velocity demand itself: $\sigma_v = \sigma_f v$, with $\sigma_f \approx 0.10$.

Naming the estimated position vector $\mathbf{f}_v$ and the control vector $\mathbf{u}$, the covariance matrix $\mathtt{Q}$ of $\mathbf{f}_v$ can be calculated as

$$\mathbf{f}_v = \begin{pmatrix} z(t+\Delta t) \\ x(t+\Delta t) \\ \phi(t+\Delta t) \end{pmatrix} \quad , \quad \mathbf{u} = \begin{pmatrix} v \\ s \end{pmatrix} \quad , \quad \mathtt{Q} = \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}} \mathtt{U} \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}}^\top \quad , \tag{1}$$

where $\mathtt{U}$ is the diagonal covariance matrix of $\mathbf{u}$.

2.2 Head Geometry

Fixating a feature with the active head provides a 3D measurement of its location from stereo and knowledge of the head odometry (Figure 1(b)). Locating a feature at position u_L, v_L in the left camera's image and u_R, v_R in the right one, its 3D position $\mathbf{h}_G$ relative the the head centre can be calculated in the $C0$ vehicle-centred frame. Using knowledge of the head's joint angles, the vectors $\mathbf{p}_L$, $\mathbf{c}_L$, $\mathbf{n}_L$ and $\mathbf{p}_R$, $\mathbf{c}_R$, $\mathbf{n}_R$ are formed and summed to obtain the vector locations of the two optic centres with respect to the head centre. $\mathbf{h}_G$ can then be expressed as either:

$$\mathbf{h}_G = \mathbf{p}_L + \mathbf{c}_L + \mathbf{n}_L + \mathbf{h}_L \quad \text{or} \quad \mathbf{h}_G = \mathbf{p}_R + \mathbf{c}_R + \mathbf{n}_R + \mathbf{h}_R \, .$$

Vectors $\mathbf{h}_L$ and $\mathbf{h}_R$ can be found up to scale in the vehicle-centred coordinate frame:

$$\mathbf{h}_L{}^{C0} \propto \mathtt{M}^{\mathtt{COL}} \mathtt{C}_L^{-1} (u_L, v_L, 1)^\mathsf{T} \quad \text{and} \quad \mathbf{h}_R{}^{C0} \propto \mathtt{M}^{\mathtt{COR}} \mathtt{C}_R^{-1} (u_R, v_R, 1)^\mathsf{T} \, ,$$

where $\mathtt{M}^{\mathtt{COL}}$ and $\mathtt{M}^{\mathtt{COR}}$ are the (known) rotation matrices transforming from the left and right camera-centred coordinate systems into the vehicle coordinate system, and $\mathtt{C}_{L,R}$ are the (known) camera calibration matrices.

The feature position relative to the head centre, $\mathbf{h}_G$, is found by back-projecting the two rays defined by $\mathbf{h}_L$ and $\mathbf{h}_R$ and finding their intersection in 3D space — in the presence of noise, the midpoint of their mutually perpedicular bisector is used.

3 The Map-Building and Localisation Algorithm

3.1 The State Vector and its Covariance

Current estimates of the locations of the vehicle and the scene features which are known about are stored in the system state vector $\hat{\mathbf{x}}$, and the uncertainty of the estimates in the covariance matrix $\mathtt{P}$. These are partitioned as follows:

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \end{pmatrix} \quad , \quad \mathtt{P} = \begin{bmatrix} \mathtt{P}_{xx} & \mathtt{P}_{xy_1} & \mathtt{P}_{xy_2} & \cdots \\ \mathtt{P}_{y_1x} & \mathtt{P}_{y_1y_1} & \mathtt{P}_{y_1y_2} & \cdots \\ \mathtt{P}_{y_2x} & \mathtt{P}_{y_2y_1} & \mathtt{P}_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \, . \tag{2}$$

$\hat{\mathbf{x}}$ has $3(n+1)$ elements, where n is the number of known features. $\mathtt{P}$ is symmetric, with size $3(n+1) \times 3(n+1)$. $\hat{\mathbf{x}}_v$ is the vehicle position estimate, and $\hat{\mathbf{y}}_i$ the estimated 3D location of the ith feature:

$$\hat{\mathbf{x}}_v = (\hat{z},\ \hat{x},\ \hat{\phi})^{\mathsf{T}}\ ,\ \ \hat{\mathbf{y}}_i = (\hat{X}_i,\ \hat{Y}_i,\ \hat{Z}_i)^{\mathsf{T}}\ .$$

3.2 Moving and Making Predictions

The robot's motion is discretised into steps of time interval Δt, with an incrementing label k affixed to each. Δt is set to be the smallest interval at which changes are made to the vehicle control inputs v and s, allowing the motion model of Section 2.1 to be used. After a step of movement, a new state estimate and covariance are produced:

$$\hat{\mathbf{x}}(k+1|k) = \begin{pmatrix} \mathbf{f}_v(\mathbf{x}_v(k|k), \mathbf{u}(k)) \\ \hat{\mathbf{y}}_1(k|k) \\ \hat{\mathbf{y}}_2(k|k) \\ \vdots \end{pmatrix} \tag{3}$$

$$\mathtt{P}(k+1|k) = \begin{bmatrix} \frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}\mathtt{P}_{xx}(k|k)\frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}^{\top} + \mathtt{Q}(k) & \frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}\mathtt{P}_{xy_1}(k|k) & \frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}\mathtt{P}_{xy_2}(k|k) & \dots \\ \mathtt{P}_{y_1x}(k|k)\frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}^{\top} & \mathtt{P}_{y_1y_1}(k|k) & \mathtt{P}_{y_1y_2}(k|k) & \dots \\ \mathtt{P}_{y_2x}(k|k)\frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}^{\top} & \mathtt{P}_{y_2y_1}(k|k) & \mathtt{P}_{y_2y_2}(k|k) & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}, \tag{4}$$

where $\mathbf{f}_v$ and $\mathtt{Q}(k)$ are as defined in Equation 1. This new covariance matrix is formulated from the usual EKF prediction rule $\mathtt{P}(k+1|k) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathtt{P}(k|k)\frac{\partial \mathbf{f}}{\partial \mathbf{x}}^{\top} + \mathtt{Q}(k)$, where $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ is the full state transition Jacobian:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v} & \mathtt{0} & \mathtt{0} & \dots \\ \mathtt{0} & \mathtt{I} & \mathtt{0} & \dots \\ \mathtt{0} & \mathtt{0} & \mathtt{I} & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}\ .$$

3.3 Updating the State Vector After a Measurement

When the location of a feature is measured, as described in Section 2.2 we obtain a measurement in the vehicle frame $C0$ of the vector $\mathbf{h}_G$ from the head centre to the feature. The function giving the predicted cartesian components of this measurement is:

$$\mathbf{h}_{Gi} = \begin{pmatrix} h_{Gix} \\ h_{Giy} \\ h_{Giz} \end{pmatrix} = \begin{pmatrix} \cos\phi(X_i - x) - \sin\phi(Z_i - z) \\ Y_i - H \\ \sin\phi(X_i - x) + \cos\phi(Z_i - z) \end{pmatrix}\ . \tag{5}$$

Before processing measurements, however, they are transformed into an angular form:

$$\mathbf{h}_i = (\alpha_i\ ,\ e_i\ ,\ \gamma_i)^{\mathsf{T}} = (\tan^{-1}\frac{h_{Gix}}{h_{Giz}}\ ,\ \tan^{-1}\frac{h_{Giy}}{h_{Gi\rho}}\ ,\ \tan^{-1}\frac{I}{2h_{Gi}})^{\mathsf{T}}\ , \tag{6}$$

where h_{Gi} is the length of vector $\mathbf{h}_{Gi}$ and $h_{Gi\rho} = \sqrt{{h_{Gix}}^2 + {h_{Giz}}^2}$ is its projection onto the xz plane. I is the inter-ocular separation of the active head. These angles represent the pan, elevation and vergence angles respectively of an ideal active head positioned at the head centre and fixating the feature, "ideal" here meaning a head that does not have the offsets that the real head has (in terms of Figure 1(b) this would mean that vectors $\mathbf{n}$ and $\mathbf{c}$ would be zero, with $\mathbf{p}$ purely along the elevation axis). Now α_i, e_i and γ_i will be very close to the actual pan, elevation and vergence angles of the real active head at fixation, but accuracy is gained by taking account of all the head offsets in this way.

The reason for using an angular measurement representation at all is that it allows measurement noise to be represented as a constant, diagonal matrix. The largest error in measurements is in the accuracy with which features can be located in the image centre — the rule used is that a successful fixation lock-on has been achived when the feature is located within a radius of two pixels from the principal point in both images. This represents an angular uncertainty of around $0.3°$, and Gaussian errors of this standard deviation are assigned to α_i, e_i and γ_i. The angular errors in measurements from the head axis encoders are much smaller than this and can be neglected. The measurement noise covariance matrix is therefore:

$$\mathtt{R} = \begin{bmatrix} \Delta\alpha^2 & 0 & 0 \\ 0 & \Delta e^2 & 0 \\ 0 & 0 & \Delta\gamma^2 \end{bmatrix} . \tag{7}$$

With a diagonal $\mathtt{R}$, measurements α_i, e_i and γ_i are independent. This has two advantages: first, potential problems with bias are removed from the filter update by representing measurements in a form where the noise can closely be modelled as Gaussian. Second, the measurement vector $\mathbf{h}_i$ can be decoupled, and scalar measurement values used to update the filter in sequence. This is computationally beneficial since it is now not necessary to invert any matrices in the update calculation. For each scalar part of the measurement h_i (where h_i is one of α_i, e_i, γ_i for the current feature of interest), the Jacobian

$$\frac{\partial h_i}{\partial \mathbf{x}} = \left(\tfrac{\partial h_i}{\partial \mathbf{x}_v}\ \mathtt{0} \ldots \mathtt{0}\ \tfrac{\partial h_i}{\partial \mathbf{y}_i}\ \mathtt{0} \ldots \right) .$$

is formed. This row matrix has non-zero elements only at locations corresponding to the state of the vehicle and the feature in question, since $h_i = h_i(\mathbf{x}_v, \mathbf{y}_i)$. The scalar innovation variance S is calculated as:

$$S = \frac{\partial h_i}{\partial \mathbf{x}} \mathtt{P} {\frac{\partial h_i}{\partial \mathbf{x}}}^\top + R = \frac{\partial h_i}{\partial \mathbf{x}_v} \mathtt{P}_{xx} {\frac{\partial h_i}{\partial \mathbf{x}_v}}^\top + 2 \frac{\partial h_i}{\partial \mathbf{x}_v} \mathtt{P}_{xy_i} {\frac{\partial h_i}{\partial \mathbf{y}_i}}^\top + \frac{\partial h_i}{\partial \mathbf{y}_i} \mathtt{P}_{y_i y_i} {\frac{\partial h_i}{\partial \mathbf{y}_i}}^\top + R , \tag{8}$$

where $\mathtt{P}_{xx}$, $\mathtt{P}_{xy_i}$ and $\mathtt{P}_{y_i y_i}$ are 3×3 blocks of the current state covariance matrix $\mathtt{P}$, and R is the scalar measurement noise variance ($\Delta\alpha^2$, Δe^2 or $\Delta\gamma^2$) of the measurement. The Kalman gain $\mathtt{W}$ can then be calculated and the filter update performed in the usual

way:

$$\mathtt{W} = \mathtt{P}\frac{\partial h_i}{\partial \mathbf{x}} S^{-1} = S^{-1} \begin{pmatrix} \mathtt{P}_{xx} \\ \mathtt{P}_{y_1 x} \\ \mathtt{P}_{y_2 x} \\ \vdots \end{pmatrix} \frac{\partial h_i}{\partial \mathbf{x}_v} + S^{-1} \begin{pmatrix} \mathtt{P}_{x y_i} \\ \mathtt{P}_{y_1 y_i} \\ \mathtt{P}_{y_2 y_i} \\ \vdots \end{pmatrix} \frac{\partial h_i}{\partial \mathbf{y}_i} \tag{9}$$

$$\hat{\mathbf{x}}_{new} = \hat{\mathbf{x}}_{old} + \mathtt{W}(z_i - h_i) \tag{10}$$

$$\mathtt{P}_{new} = \mathtt{P}_{old} - \mathtt{W} S \mathtt{W}^\top . \tag{11}$$

z_i is the actual measurement of the quantity obtained from the head, and h_i is the prediction. This update is carried out sequentially for each scalar element of the measurement.

3.4 Initialising a New Feature

When an unknown feature is observed for the first time, a vector measurement $\mathbf{h}_G$ is obtained of its position relative to the head centre, and its state is initialised to

$$\mathbf{y}_i = \begin{pmatrix} x + h_{Gix}\cos\phi + h_{Giz}\sin\phi \\ H + h_{Giy} \\ z - h_{Gix}\sin\phi + h_{Giz}\cos\phi \end{pmatrix} . \tag{12}$$

Jacobians $\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}$ and $\frac{\partial \mathbf{y}_i}{\partial \mathbf{h}_G}$ are calculated and used to update the total state vector and covariance (assuming for example's sake that two features are known and the new one becomes the third):

$$\mathbf{x}_{new} = \begin{pmatrix} \mathbf{x}_v \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_i \end{pmatrix} \tag{13}$$

$$\mathtt{P}_{new} = \begin{bmatrix} \mathtt{P}_{xx} & \mathtt{P}_{xy_1} & \mathtt{P}_{xy_2} & \mathtt{P}_{xx}\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}^\top \\ \mathtt{P}_{y_1 x} & \mathtt{P}_{y_1 y_1} & \mathtt{P}_{y_1 y_2} & \mathtt{P}_{y_1 x}\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}^\top \\ \mathtt{P}_{y_2 x} & \mathtt{P}_{y_2 y_1} & \mathtt{P}_{y_2 y_2} & \mathtt{P}_{y_2 x}\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}^\top \\ \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}\mathtt{P}_{xx} & \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}\mathtt{P}_{xy_1} & \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}\mathtt{P}_{xy_2} & \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}\mathtt{P}_{xx}\frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_v}^\top + \frac{\partial \mathbf{y}_i}{\partial \mathbf{h}_G}\mathtt{R}_L\frac{\partial \mathbf{y}_i}{\partial \mathbf{h}_G}^\top \end{bmatrix} \tag{14}$$

where $\mathtt{R}_L$ is the measurement noise $\mathtt{R}$ transformed into cartesian measurement space.

3.5 Deleting a Feature

A similar Jacobian calculation shows that that deleting a feature from the state vector and covariance matrix is a simple case of removing the rows and columns which contain it. An example in a system where the second of three known features is deleted:

$$\begin{pmatrix} \mathbf{x}_v \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{x}_v \\ \mathbf{y}_1 \\ \mathbf{y}_3 \end{pmatrix} \quad , \quad \begin{bmatrix} \mathtt{P}_{xx} & \mathtt{P}_{xy_1} & \mathtt{P}_{xy_2} & \mathtt{P}_{xy_3} \\ \mathtt{P}_{y_1 x} & \mathtt{P}_{y_1 y_1} & \mathtt{P}_{y_1 y_2} & \mathtt{P}_{y_1 y_3} \\ \mathtt{P}_{y_2 x} & \mathtt{P}_{y_2 y_1} & \mathtt{P}_{y_2 y_2} & \mathtt{P}_{y_2 y_3} \\ \mathtt{P}_{y_3 x} & \mathtt{P}_{y_3 y_1} & \mathtt{P}_{y_3 y_2} & \mathtt{P}_{y_3 y_3} \end{bmatrix} \rightarrow \begin{bmatrix} \mathtt{P}_{xx} & \mathtt{P}_{xy_1} & \mathtt{P}_{xy_3} \\ \mathtt{P}_{y_1 x} & \mathtt{P}_{y_1 y_1} & \mathtt{P}_{y_1 y_3} \\ \mathtt{P}_{y_3 x} & \mathtt{P}_{y_3 y_1} & \mathtt{P}_{y_3 y_3} \end{bmatrix} . \tag{15}$$

3.6 Zeroing the Coordinate Frame

As mentioned in the introduction, it is possible to re-zero the world coordinate frame at the current vehicle position. The new state becomes:

$$\mathbf{x}_{new} = \begin{pmatrix} \mathbf{x}_{vnew} \\ \mathbf{y}_{1new} \\ \mathbf{y}_{2new} \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{h}_{G1} + \mathbf{H} \\ \mathbf{h}_{G2} + \mathbf{H} \\ \vdots \end{pmatrix}, \tag{16}$$

where $\mathbf{h}_{Gi}$ is the current vector from the head centre to feature i as given in Equation 5, and $\mathbf{H}$ is the constant vector describing the vertical offset from the ground plane to the head centre. To calculate the new state covariance we form the sparse Jacobian matrix:

$$\frac{\partial \mathbf{x}_{new}}{\partial \mathbf{x}_{old}} = \begin{bmatrix} 0 & 0 & 0 & \dots \\ \frac{\partial \mathbf{h}_{G1}}{\partial \mathbf{x}_v} & \frac{\partial \mathbf{h}_{G1}}{\partial \mathbf{y}_1} & 0 & \dots \\ \frac{\partial \mathbf{h}_{G2}}{\partial \mathbf{x}_v} & 0 & \frac{\partial \mathbf{h}_{G2}}{\partial \mathbf{y}_2} & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}, \tag{17}$$

and calculate $\mathtt{P}_{new} = \frac{\partial \mathbf{x}_{new}}{\partial \mathbf{x}_{old}} \mathtt{P}_{old} \frac{\partial \mathbf{x}_{new}}{\partial \mathbf{x}_{old}}^{\top}$.

4 Implementation

All visual processing and the localisation filter are implemented in C++ on a 100 MHz Pentium PC operating under Linux. The PC hosts a Matrox Meteor for stereo image capture, a Delta-Tau PMAC controller to direct the head, and a proprietary Transputer controller to drive the vehicle.

As the robot drives at speeds of up to 20cms^{-1}, a feature may be tracked at fixation at a frequency of 5Hz: the system predicts, moves the active head to fixation, obtains a measurement and incorporates it into the filter in a time less than 0.2s. Alternatively, the robot may stop and make sequential measurements of several features. The main factor limiting speed of operation is the time required to carry out expensive correlation searches (see Section 5.2) on this general purpose hardware. If the robot's position is very uncertain (possibly after failed measurements), search regions become large, search times long, and the robot velocity and the frequency of measurements are automatically reduced accordingly.

Maintaining the large state vector and covariance becomes comparably computationally costly only with a large number ($\approx$ 25) of features. We have developed a method (not described here) whereby execution speed is not compromised by this, even with very large numbers of features, by postponing the full, but exact, update of the whole state until the robot has stopped moving and has some unutilised processing time.

5 Features

5.1 Detecting and Initialising Features

Visual landmarks should be stationary, point features which are easily distinguishable from their surroundings. In this work we use the operator of Shi and Tomasi [8] to

selects regions of high interest, and represent the features as 15×15 pixel patches. Typical features found in this way are shown in Figures 2(a) and Figure 5.

To initialise a new feature, the patch operator is applied in the left camera image. For the best patch found, an epipolar line is generated in the right image (using knowledge of the head geometry), and the nearby region searched. If a good stereo match is found, the two pairs of image coordinates (u_L, v_L) and (u_R, v_R) allow the feature's 3D location in the vehicle-centred coordinate frame to be calculated. The head is driven to fixate the feature, (with symmetric left and right head vergence angles enforced to remove redundancy), and re-measured, being easily found now near the centre of the images. Making all measurements at fixation reduces the need for accurate knowledge of the camera focal lengths. The feature is then initialised in the map as detailed in Section 3.4.

Typical indoor environments provide many features which are suitable for tracking, as well as some which, because of partial occlusion or reflection, are not. While no attempt is made to discern these bad features at the initialisation stage, they can be rejected later because they will not be tracked under the constraints of the filter (see Section 5.3).

5.2 Measuring and Tracking Features

The Kalman Filter approach means that a prediction is available of any measurement to be made, and a prediction covariance. When measuring a known feature with the active head, use of the prediction is essential to drive the cameras to the expected fixation angles to make the feature visible. The prediction covariance is then used to produce search areas in the two images within which the feature must lie with a high probability.

Having calculated $\mathbf{h}_{Gi}$, the predicted vector from the head centre to the feature, as in Equation 5, along with its covariance, the system drives the head to fixation and calculates in camera centred coordinates the vectors $\mathbf{h}_L$ and $\mathbf{h}_R$ from the camera optic centres to the feature and their covariances $\mathtt{P}_{h_L}$ and $\mathtt{P}_{h_R}$. Both of these vectors will have zero x and y components since at fixation the z-axis-defining optic axes pass through the feature. Considering the left camera, image projection is defined by the the usual equations:

$$u_L = -fk_u(h_{Lx}/h_{Lz}) + u_0 \quad \text{and} \quad v_L = -fk_v(h_{Ly}/h_{Lz}) + v_0 \,. \tag{18}$$

The covariance matrix of the image vector $\mathbf{u}_L = (u_L\, v_L)^\top$ is given by $\mathtt{U}_L = \frac{\partial \mathbf{u}_L}{\partial \mathbf{h}_L} \mathtt{P}_{h_L} \frac{\partial \mathbf{u}_L}{\partial \mathbf{h}_L}^\top$. The value of the Jacobian at $h_{Lx} = h_{Ly} = 0$ is

$$\frac{\partial \mathbf{u}_L}{\partial \mathbf{h}_L} = \begin{bmatrix} \frac{-fk_u}{h_{Lz}} & 0 & 0 \\ 0 & \frac{-fk_v}{h_{Lz}} & 0 \end{bmatrix} .$$

Specifying a number of standard deviations (typically 3), $\mathtt{U}_L$ defines an ellipse in the left image which is searched for the feature patch using normalized cross-correlation The same procedure is followed in the right image. Limiting our search for feature patches to these areas not only maximises computational efficiency but also minimises the chance of obtaining mismatches. Figure 2(b,c) shows examples of the elliptical search regions, and (d-g) shows a feature tracked by the head over a large robot motion.

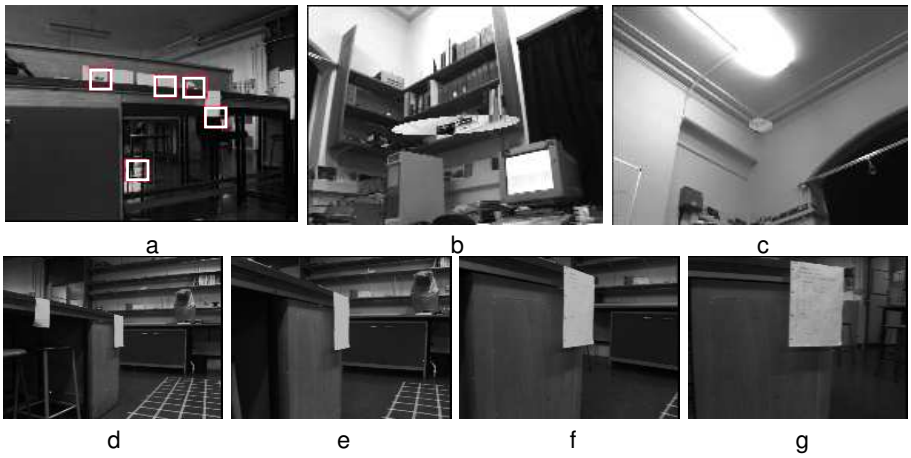

Fig. 2. (a) Typical features detected using Shi and Tomasi. (b,c) Examples of image search ellipses with large and small uncertainty. (d-g) Continuous fixation of a feature by the head.

5.3 Selecting Between Features and Maintaining the Map

As discussed in the introduction, when moving through an unfamiliar world the visual system needs to perform the joint roles of building a useful map and telling the robot where it is on that map. A strategy is needed to direct attention to where it is most needed, since during its motion the robot is able to track just one feature at a time. Three issues need to be considered, viz:
1. Which of the current set of known features should be tracked?
2. Is it time to label any features as not useful and abandon them?
3. Is it time to look for new features?

Since the features are simple image patches, they are not expected to be recognisable from all viewpoints. The expected visibility of a feature is defined based on the difference in angle and distance between the robot's current estimated viewpoint and the viewpoint from which it first identified the feature (angle differences of up to $45°$ and distance ratios of up to 1.4 being tolerated). Using different types of feature, different visibility criteria would be appropriate (we are currently investigating using 3D planar patches as features).

In answer to Question 3 then, the robot uses the heuristic that new features should be found if less than two are currently visible. The system attempts to initialise three widely-spaced new features by examining regions to the left, right and straight ahead. More purposive approaches to finding new features in optimal positions have been considered, but in most environments the critical factor is finding reliable features at all. There may not be anything to see in an "optimal" position. Once features have been found, an intelligent choice can be made about which is best to observe from an information point-of-view. To answer Question 2, a feature is abandoned if, after at least 10 attempted measurements from viewpoints from which it should be visible, more than half have failed. Features not corresponding to true point features, having very

low viewpoint-invariance, or which are frequently occluded, are quickly rejected and deleted from the map.

To tackle Question 1, the principle of making a measurement where the ability to predict is least, as discussed in recent work by Whaite and Ferrie [9], is used. Given a choice of measurements in a system where the uncertainty in estimates of the parameters of interest is known, it makes sense to make the one where we are least certain of the result, since this will in a sense "squash" the total uncertainty, viewed as a multi-dimensional ellipsoid, along the longest axis available.

Whenever the robot is to make a measurement of a feature, a predicted measurement $\mathbf{h}$ is formed and the innovation covariance matrix $\mathtt{S}$ is calculated. This matrix describes how much the actual measurement is expected to vary from the prediction. To produce scalar numbers to use as the basis for decisions about which feature to observe, the volume V_S in (α, e, γ) space of the ellipsoid represented by $\mathtt{S}$ at the 3σ level can be calculated for each visible feature. That having the highest V_S is chosen for tracking. The highest possible integrity in the whole map / robot estimation is retained in this way.

The most striking result of this criterion seen in experiments is that it demands frequent changes of tracked feature. Once one or two measurements have been made of a feature, the criterion tells us that there is not much more information to be gleaned from it at the current time, and it is best to switch attention to another. This is a result of the way that tracking one point feature, even with perfect measurements, does not fully constrain the robot's motion — uncertainty is always growing in some direction.

6 Experiments

6.1 Comparison with Ground Truth

To evaluate the localisation and map-building accuracy of the system, a corridor-like area of a large laboratory was laid with a grid and a set of features in known positions was set up in a regularly-spaced line (see Figure 3). Starting from the grid origin, the robot was driven forward in a nominally straight line. Every second feature in the line was initialised and tracked for a short while on this outward journey, the robot stopping at frequent intervals so that ground-truth measurements could be made of its position relative to the grid, and orientation using an on-board laser pointer. The robot then reversed back down the corridor, tracking the features it had *not* previously seen. Once it had returned to near its starting position, it drove forward again, now attempting to re-measure features found early on, thus completing the loop on the motion and establishing that a reliable map had been formed. It will be shown how much better the single filter approach performs than the multiple-filters methods seen in the literature.

Results directly generated by the full system are shown in Figure 4 superimposed on the measured ground truth. The robot's position was tracked well by the filter as it moved forward and back, but it can be seen that drift started to occur; in particular by the 4th picture, when the robot was close again to its starting position, the filter estimated the robot's position as $z = -0.11$m, $x = 0.26$m, $\phi = -0.08$rad when the true position was $z = 0.01$m, $x = 0.02$m, $\phi = 0.02$rad. This discrepancy is to be expected, since the robot had continually been tracking new features without referring to previously known ones (motion drift). However, the filter had correctly montitored

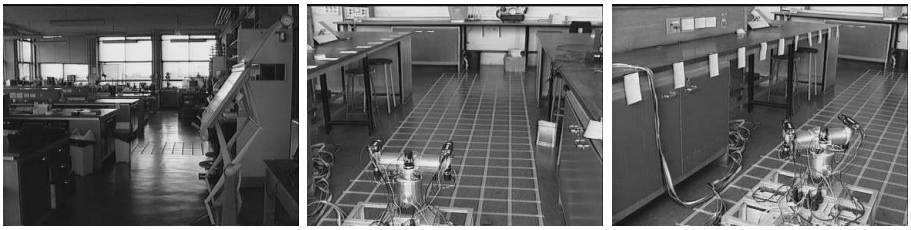

Fig. 3. (a) The laboratory used in experiments. (b) The corridor-like bay where ground-truth measurements could be referenced to a 20cm grid (not used by the vision system). (c) The set of regularly-spaced reference features used in the ground-truth experiment.

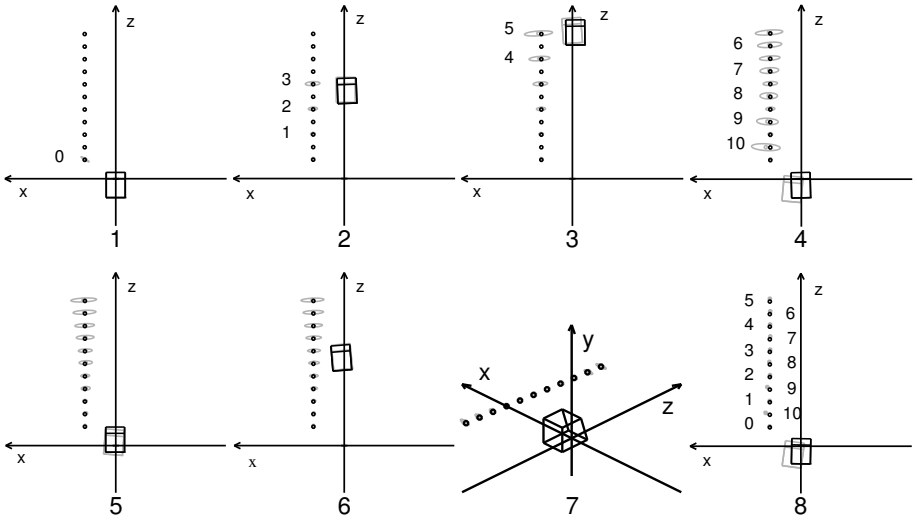

Fig. 4. Experiment with regularly-spaced landmarks: estimated positions of the robot ($\hat{\mathbf{x}}_v$) and features ($\hat{\mathbf{y}}_i$) in grey, along with 3σ ellipses for the point covariances $\mathtt{P}y_iy_i$, are shown superimposed on the true positions (from manual measurement) in black as the robot moved forward and back. The feature spacing was 40cm, and the robot moved about 5m from its origin. Feature labels show the order they were tracked in.

this uncertainty: in the next step (5th picture), measurements had been made of feature 0. This, the very first feature measured, had a position which was very well known in the world coordinate frame since the robot's position had been very certain at this time. Making these measurements therefore locked down the vehicle uncertainty and a much better estimate was produced: estimated position $z = 0.31$m, $x = 0.04$m, $\phi = -0.08$rad, true position $z = 0.39$m, $x = 0.02$m, $\phi = 0.00$rad. The covariance matrices of the robot state before and after the measurements were:

$$\mathtt{P}_{xx}(4) = \begin{bmatrix} 0.0039 & -0.0095 & 0.0036 \\ -0.0095 & 0.0461 & -0.0134 \\ 0.0036 & -0.0134 & 0.0051 \end{bmatrix} \quad , \quad \mathtt{P}_{xx}(5) = \begin{bmatrix} 0.0016 & -0.0004 & 0.0016 \\ -0.0004 & 0.0002 & -0.0004 \\ 0.0016 & -0.0004 & 0.0018 \end{bmatrix}$$

The estimates of the locations of all the other features were also immediately improved. The remaining discrepancy in the z and ϕ estimates reflects the fact that measuring one feature does not fully constrain the robot location: it can be seen that these estimates improved when the robot had re-measured more features by the 6th picture. The 7th picture shows the state from the 6th picture zeroed with respect to the robot as in Section 3.6 and displayed at an angle to show the three-dimensionality of the map generated.

In the 8th picture, results are shown from a repeat of the experiment demonstrating the failure of methods using separate filters for the robot and feature states to achieve re-registration with original features. With our implementation, these methods are simulated by simply zeroing all off-diagonal blocks of the total covariance matrix `P` after each prediction and update (this can be shown to be equivalent to the most sensible approach using multiple filters). The picture shows the state once the robot had returned to near its starting position, and it can be seen that while similar drift was observed in the robot and feature estimates as above, correct account was not taken of this and the covariances were underestimated (the feature uncertainty ellipses being too small to see). An attempt to re-measure feature 0 from this position failed because it did not lie within the image search ellipse generated. The method has no ability to recover from this situation.

6.2 A Fully Automatic Experiment

We present results from a fully automatic run where the robot drove up and down the corridor-like bay with no initial knowledge of the world. The instructions given to the robot were to head in sequence from its starting point at $(z, x) = (0, 0)$ to the waypoints $(6, 0.4)$, $(6, 0)$, and finally $(0, 0)$ again (in metre units). When heading for a particular waypoint, the robot continuously controls its steering angle s according to a simple law described in [6]. A particular waypoint is said to have been reached when the robot's estimated position is within 30cm of it, and the robot starts to steer for the next one.

The robot's progress is shown in frames cut from a video in Figure 5, along with saved views of the first 16 features detected and initialised into the map as in Section 5.3. The output of the filter appears in Figure 6, where those features are annotated. Some of these features did not survive very long before being abandoned as not useful (numbers 4 and 5 in particular not surviving past very early measurement attempts and not being displayed in Figure 6). Others, such as 0, 12 and 14 proved to be very durable, being easy to see and match from all positions from which they are expected to be visible. It can be seen that many of the best features found lie near the ends of the corridor, particularly the large number found near the cluttered back wall (11–15, etc.). The active approach really comes into its own during sharp turns such as that being carried out in the 5th picture, where features such as these could be tracked continuously, using the full range of movement of the neck axis, while the robot made a turn of $180°$. The angle of turn can be estimated accurately at a time when odometry data is unreliable.

Outward Journey: the sequence of features selected to be tracked in the early stages of the run (up to the 3rd picture in Figure 6)) was 0, 2, 1, 0, 2, 1, 3, 5, 4, 7, 6, 8, 3, 6, 8, 7, 3, 7, 8, 3, 9 — we see frequent switching between a certain set of features until some go out of visibility and it is necessary to find new ones.

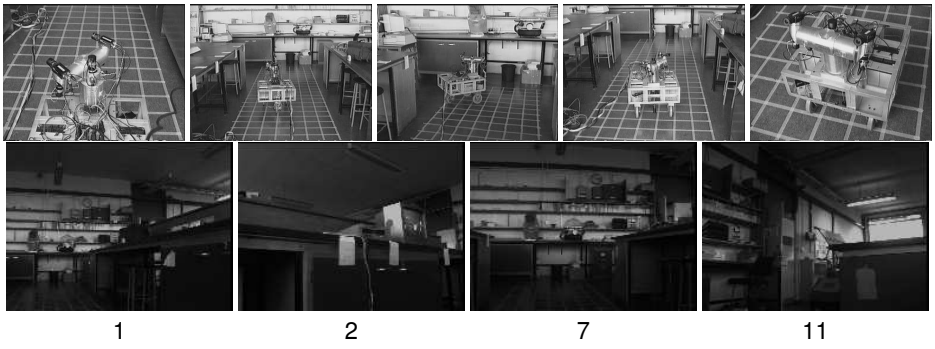

Fig. 5. The robot navigating autonomously up and down the corridor. The lower four images are example fixated views of four of the features initialised as landmarks. Their numbering corresponds with that used in Figure 6.

Return to Origin: in the 6th picture of Figure 6, the robot had reached its goal, the final waypoint being a return to its starting point. The robot had successfully refound original features on its return journey, in particular feature 0 whose position was very well known. The choice of feature criterion described in Section 5.3 had demanded re-measurement of these features as soon as they became visible again, reflecting the drift occurring between the robot position estimate and the world coordinate frame. The robot's true position relative to the grid was measured here, being $z = 0.06$m, $x = -0.12$m, $\phi = 3.05$rad, compared to the estimated position $z = 0.15$m, $x = -0.03$m, $\phi = 2.99$rad.

Repeat Journey: the experiment was continued by commanding the robot back out to $(z,\ x)$ = (6, 0), then home again to (0, 0). In these further runs, the system needed to do far less map-building since a large number of features was already known about along the trajectory. At (6, 0), shown in the 7th picture, the robot's true position was $z = 5.68$m, $x = 0.12$m, $\phi = 0.02$rad, and estimated state was $z = 5.83$m, $x = 0.00$m, $\phi = -0.02$rad. At (0, 0) again, shown in the 8th picture, the true position of $z = 0.17$m, $x = -0.07$m, $\phi = -3.03$rad compared with the estimate $z = 0.18$m, $x = 0.00$m, $\phi = -3.06$rad. This is very impressive localisation performance considering that the robot had travelled a total distance of some 24m by this stage.

6.3 Incorporating Prior Map Knowledge

In a real application of an autonomous robot, it is likely that some prior knowledge about the world will be provided, or indeed that this will be necessary to the robot's ability to perform a useful task. In particular, our system as described does not give the robot the capability to detect obstacles along a requested route. It is not intended that the map of features used as landmarks be dense enough to use as a basis for obstacle avoidance.

Prior knowledge of a feature can be incorporated into the map by initialising its location into the state vector at the start of motion, and providing the system with the feature description (i.e. an image patch). If the feature location is precisely known, this

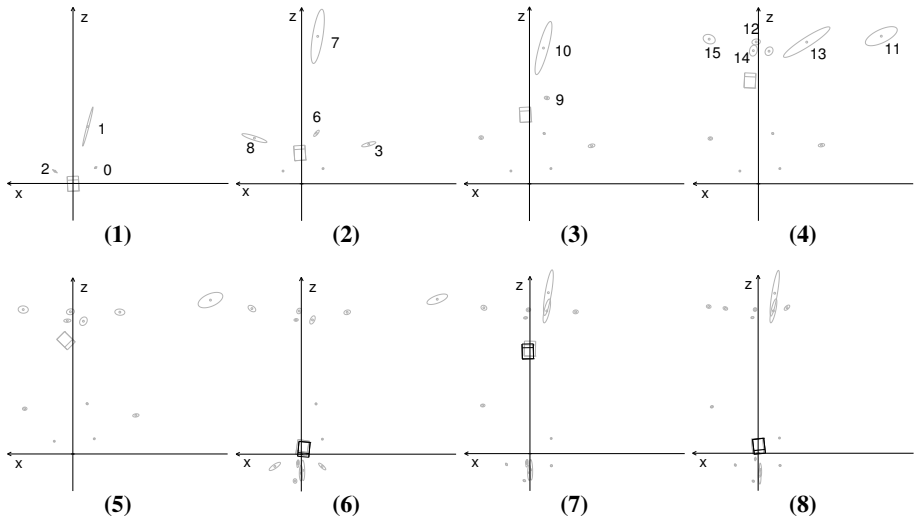

Fig. 6. The map built in autonomous navigation up and down a corridor. Grey shows the estimated locations of the vehicle and features, and black (where measured) the true vehicle position. The furthest features lie at $z \approx 8$m.

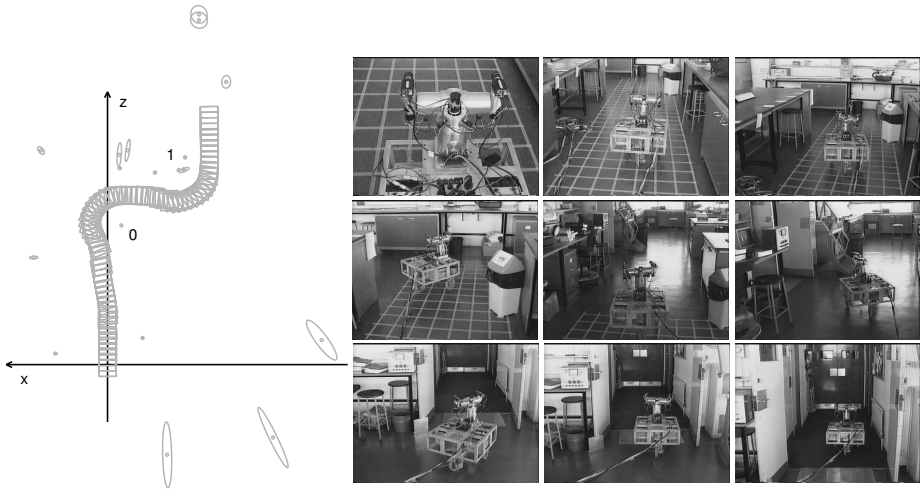

Fig. 7. The estimated trajectory and frames cut from a video as the robot navigated autonomously around two known landmarks and out of the laboratory door.

is correctly managed by the filter by simply setting all covariance elements relating to that feature to zero — otherwise, some non-zero measurement uncertainty could be initialised. No special treatment needs to be accorded to the feature after this. Having perfect features in the scene means that the robot is able to remain true to the world coordinate frame over a wider area than otherwise.

In addition, extra labels can be attached to features initialised in this way to aid navigation, as we demonstrate in experiment here. The locations of two features lying on the corners of a zig-zag path were given to the robot as prior knowledge, along with instructions to steer to the left of the first and to the right of the second on its way to a final location at $(z, x) = (9, \ -3.5)$. This is information that could be assigned to automatically detected features by additional visual modules such as a free-space detector.

Figure 7 shows the locations of the known features 0 and 1, the map of other features which was formed, and the estimated robot trajectory as it negotiated the corners to pass out of the laboratory door. Steering around the known features was accomplished with a similar steering law to that used to steer to a waypoint, but with the aim of avoidance by a safe radius of 1m [6]. It can be seen that features detected in the vicinity of the known ones (especially 1) are also well known. Small "kinks" in the trajectory are noticeable where the robot first made successful measurements of the known features and made relatively large re-registration adjustments to its position estimates.

7 Conclusions

This paper has shown how a robot can use active vision to provide continuous and accurate global localisation information by serially fixating its cameras on arbitrary features in the scene. A map of landmarks is automatically built and maintained, and extended periods of navigation are permitted by the robot's ability to identify the same features again and again, mitigating the problem of motion drift.

The system provides a framework into which goal-directed visual capabilities can be inserted, as demonstrated in experiments. In future work, it is planned to add modules which enable a wide variety of purposive manoeuvres, such as automatic free-space detection to permit obstacle avoidance, or target recognition to provide goals automatically.

Potential improvements to the localisation system itself are chiefly concerned with allowing different sorts of scene feature to be used as landmarks: the additional use of line segments or planar patch features would extend robustness by making reliable landmarks easy to find and track in any environment.

References

1. N. Ayache. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*. MIT Press, Cambridge MA, 1991.
2. P. A. Beardsley, I. D. Reid, A. Zisserman, and D. W. Murray. Active visual navigation using non-metric structure. In *Proc 5th Int Conf on Computer Vision, Boston MA*, pages 58–65. IEEE Computer Society Press, 1995.
3. J.-Y. Bouget and P. Perona. Visual navigation using a single camera. In *Proc 5th Int Conf on Computer Vision, Boston MA*, pages 645–652. IEEE Computer Society Press.
4. C. G. Harris and J. M. Pike. 3D positional integration from image sequences. In *Proc 3rd Alvey Vision Conf, Cambridge UK*, pages 233–236, 1987.
5. J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Navigation*. Kluwer Academic Press, 1992.

6. D. W. Murray, I. D. Reid, and A. J. Davison. Steering and navigation behaviours using fixation. In *Proc 7th British Machine Vision Conf, Edinburgh*, pages 635–644, 1996.
7. P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan. A modular head/eye platform for real-time reactive vision. *Mechatronics*, 3(4):517–535, 1993.
8. Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
9. P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, 1997.