Simultaneous Localization and Map-Building Using Active Vision

Andrew J. Davison and David W. Murray, *Member*, *IEEE*

Abstract—An active approach to sensing can provide the focused measurement capability over a wide field of view which allows correctly formulated Simultaneous Localization and Map-Building (SLAM) to be implemented with vision, permitting repeatable long-term localization using only naturally occurring, automatically-detected features. In this paper, we present the first example of a general system for autonomous localization using active vision, enabled here by a high-performance stereo head, addressing such issues as uncertainty-based measurement selection, automatic map-maintenance, and goal-directed steering. We present varied real-time experiments in a complex environment.

Index Terms—Active vision, simultaneous localization and map-building, mobile robots.

1 Introduction

Incremental building and maintaining of maps for immediate use by a navigating robot has been shown to rely on detailed knowledge of the cross-coupling between running estimates of the locations of robot and mapped features [1]. Without this information, features, which are redetected after a period of neglect, are treated as new and the entire structure suffers progressive error accumulation which depends on the distance traveled, not on distance from the starting position in the fiducial coordinate frame. It becomes impossible to build persistent maps for long-term use as apparent in earlier navigation research [2], [3], [4], [5], [6], [7]. For example, Fig. 5a of reference [7], shows that the start and end of an actually closed path are recovered as different locations.

Storing and maintaining coupling information proves to be computationally expensive, in turn imposing the need to use only a sparse sets of features. This runs counter to the emphasis of recent research into visual reconstruction where large numbers of features over many images are used in batch mode to obtain accurate, dense, and visually realistic reconstructions for multimedia applications rather than robotic tasks (e.g., [8], [9]). Although batch methods provide the most accurate and robust reconstructions, the volume of calculation required for each camera location grows depending on the total length of the trajectory. Real-time applications, on the other hand, require updates to be calculable in a time bounded by a constant step interval: It is satisfying this crucial constraint which permits all-important *interaction* with the map data as it is acquired.

So, although visual sensing is the most information-rich modality for navigation in everyday environments, recent advances in simultaneous localization and map building (SLAM) for mobile robots have been made using sonar and laser range sensing to build maps in 2D and have been largely overlooked in the vision literature. Durrant-Whyte and colleagues (e.g., [10]) have implemented systems using a wide range of vehicles and sensor types and are currently working on ways to ease the computational burden of SLAM. Chong and Kleeman [11] achieved impressive results using advanced tracking sonar and accurate odometry combined with a submapping strategy. Thrun et al. [12] have produced some of the best known demonstrations of robot navigation in real environments (for example, in a museum) using laser range-finders and some vision. Castellanos [13] also used a laser range finder and a mapping strategy called the SPmap. Leonard and Feder [14], working primarily with underwater robots and sonar sensors, have recently proposed new submapping ideas, breaking a large area into smaller regions for more efficient map-building.

In this paper, we describe the first application of active vision to real-time, sequential map-building within a SLAM framework, building on our earlier work reported in [15]. We show that active visual sensing is ideally suited to the exploitation of sparse "landmark" information required in robot map-building. Using cameras with the ability both to fixate and to change fixation over a wide angular range ensures that persistent features redetected after lengthy neglect can also be *rematched*, even if the area is passed through along a different trajectory or in a different direction. This is key to reducing the effect of motion drift from the fiducial coordinate frame: The drift now depends on the distance from the origin, not the total distance traveled.

No doubt, active sensing will be implemented electronically by choosing to process only a subwindow from high-resolution omni-directional data. At present, however, full resolution multiple sensor cameras (fly-eyes) are expensive to construct and mosaicing still a research problem. On the other hand, fish-eye lenses and catadioptric mirrors [16] have the disadvantage of variable and sometimes low angular resolution. In this work, we use a agile electro-mechanical stereo head with known forward kinematics, four degrees of movement freedom, and full odometry permitting the locations of the cameras with respect to the robot to be known accurately at all times and their location to be controlled in an closed-loop sense. While an active head combines a wide field of view with high

• *The authors are with the Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, United Kingdom. E-mail: {ajd, dwm}@robots.ox.ac.uk.*

Manuscript received 9 July 1998; revised 15 June 2001; accepted 5 Dec. 2001. Recommended for acceptance by H. Christensen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107845.

sensing resolution, it also introduces the interesting penalty that a finite time is required to refixate the camera, time in which further measurements might have been made of the previously fixated scene point.

Selective sensing is the essence of the active approach and, in map-building, there is much more to be gained by making observations of some parts of the robot's surroundings than others: the two appear well-matched. Here, we only consider how active vision can provide a robot with accurate localization; but, this could be just one part of a robot's overall task. In [17], one of us described a system where attention is divided between localization and inspection. Regardless of the simplicity or complexity of the task, a rigorous statistical framework is necessary if prudent serial selection of fixation point is to be made. Although the computational complexity is high (in EKF-based SLAM, proportional to N^2, where N is the number of mapped features), real-time implementation is feasible on modest hardware, even without the various SLAM shortcut methods which have recently appeared [14], [18], [10].

The rest of the paper is organized as follows: In Section 2, we introduce the SLAM problem and discuss some of the points relevant to our implementation. We present the image processing approach and active head control strategies involved in identifying and locating natural scene features in Section 3 and Section 4 describes an experiment using contrived scene features to verify localization and map-building performance against ground-truth. We continue in Section 5 by discussing the additional sensing and processing tools, in particular, active feature selection, which are necessary in fully autonomous navigation and, in Section 6, give results from a fully automatic experiment. In Section 7, we look at supplementing SLAM with a small amount of prior knowledge and, in Section 8, bring all these elements together in a final experiment in goal-directed navigation.

2 SLAM Using Active Vision

Sequential localization and map-build based on the extended Kalman Filter (EKF) is now increasingly well understood [1], [13], [11], [19], [10] and, in this section, we only wish to establish some background and notation. Detailed expressions for the kinematics of our particular vehicle and active head can be found in [15].

2.1 The State Vector and Its Covariance

In order for information from motion models, vision, and other sensors to be combined to produce reliable estimates, sequential localization and map-building [20] involves the propagation through time of probability density functions (PDF's) representing not only uncertain estimates of the position of the robot and mapped features individually, but coupling information on how these estimates relate to each other.

The approach taken in this paper and in most other work on SLAM is to propagate first-order approximations to these probability distributions in the framework of the EKF, implicitly assuming that all PDF's are Gaussian in shape. Geometrical nonlinearity in the motion and measurement processes in most SLAM applications means that this assumption is a poor one, but the EKF has been widely demonstrated not to be badly affected by these problems. More significant is the EKF's inability to represent the multimodal PDF's resulting from imperfect data association (mismatches). The particle filtering approaches which have recently come to the fore in visual tracking research offer a solution to these problems, but in their current form are inapplicable to the SLAM problem due to their huge growth in computational complexity with state dimension [21]—in SLAM, the state consists of coupled estimates of the positions of a robot and many features and it is impossible to span a space of this state-dimension with a number of particles which would be manageable in real-time; however, some authors [22] are investigating the use of particle filters in robot localization.

In the first-order uncertainty propagation framework, the overall "state" of the system x is a vector which can be partitioned into the state $\hat{\mathrm{x}}_v$ of the robot and the states $\hat{\mathrm{y}}_i$ of entries in the map of its surroundings. The state vector is accompanied by a covariance matrix P, which can also be partitioned as follows:

$$\hat{\mathrm{x}} = \begin{pmatrix} \hat{\mathrm{x}}_v \\ \hat{\mathrm{y}}_1 \\ \hat{\mathrm{y}}_2 \\ \vdots \end{pmatrix}, \quad \mathrm{P} = \begin{pmatrix} \mathrm{P}_{xx} & \mathrm{P}_{xy1} & \mathrm{P}_{xy2} & \cdot\cdot \\ \mathrm{P}_{y1x} & \mathrm{P}_{y1y1} & \mathrm{P}_{y1y2} & \cdot\cdot \\ \mathrm{P}_{y2x} & \mathrm{P}_{y2y1} & \mathrm{P}_{y2y2} & \cdot\cdot \\ \vdots & \vdots & \vdots & \end{pmatrix}.$$

In this paper, the robot state is just ground plane position and orientation $\hat{\mathrm{x}}_v = (\hat{z}, \hat{\mathrm{x}}, \hat{\phi})^\top$ and each feature state is a 3D position $\hat{\mathrm{y}}_i = (\hat{X}_i, \hat{Y}_i, \hat{Z}_i)^\top$, but a state vector is not limited to pure position estimates: Other feature and robot attributes (such as velocity or the positions of redundant joints) can be included (e.g., [17]).

2.2 Coordinate Frames and Initialization

When the robot moves in surroundings which are initially unknown, the choice of world coordinate frame is arbitrary: Only relative locations are significant. Indeed, it is possible to do away with a world coordinate frame altogether and estimate the locations of features in a frame fixed to the robot: Motions of the robot simply appear as backwards motion of features. However, in most applications, there will be interaction with information from other frames of reference—often in the form of known way-points through which the robot is required to move (even in a case so simple as that in which the robot must return to its starting position). A world coordinate frame is essential to interact with information of this kind and, as there is little computational penalty in including an explicit robot state, we always do so (Fig. 1a).

In typical navigation scenarios (such as that of the experiments of Sections 4 and 6) where there is no prior knowledge of the environment, the world coordinate frame can be defined with its origin at the robot's starting position and the initial uncertainty relating to the robot's position in P_{xx} is zeroed.

If there is prior knowledge of some feature locations (as in the experiment of Section 7), these can be inserted explicitly into the map at initialization and this information will effectively define the world coordinate frame. The robot's starting position relative to these features must also be input and both robot and feature positions assigned suitable initial covariance values.

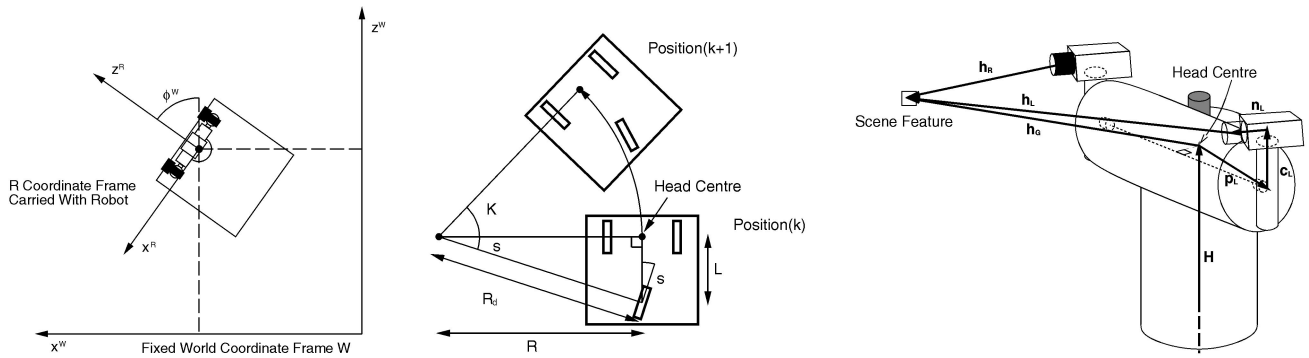

Fig. 1. (a) Coordinate frames. The robot's location in the world coordinate frame is specified by coordinates (z, x, ϕ). (b) Motion model. The vehicle's motion geometry. (c) Atcyive head model. Head geometry: The head center is at height H vertically above the ground plane.

2.3 Process Model

Temporal updating using an EKF requires prediction of the state and covariance after a robot movement during a possibly variable period Δt_k.

$$\hat{\mathrm{x}}_{v(k+1|k)} = \hat{\mathrm{f}}_v(\hat{\mathrm{x}}_{v(k|k)}, \mathbf{u}_k, \Delta t_k)$$
$$\hat{\mathrm{y}}_{i(k+1|k)} = \hat{\mathrm{y}}_{i(k|k)}, \forall i$$
$$\mathrm{P}_{(k+1|k)} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathrm{P}_{(k|k)} \frac{\partial \mathbf{f}}{\partial \mathbf{x}}^{\top} + \mathrm{Q}_k.$$

Here, $\mathbf{f}_v$ is a function of the current robot state estimate, the period, and control inputs $\mathbf{u}$, which for our robot are steering angle and wheel velocity (Fig. 1b). The robot's motion in each time step is modeled as a circular trajectory with radius R determined by wheel geometry and steering angle (see [19] for details). The full state transition Jacobian is denoted $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ and Q_k is the process noise,

$$\mathrm{Q}_k = \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}} \mathrm{U} \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}}^{\top},$$

where U is the diagonal covariance matrix of $\mathbf{u}$. Process noise accounts essentially for unmodeled effects in the vehicle motion such as wheel slippage.

2.4 Measurement Model

Any sensor which is able to measure the location of fixed features in the scene relative to the robot can contribute localization information and it is wise in implementation to separate the details of the sensors (and indeed the robot) from the algorithms used to build and process maps [20].

The key to our active approach is the ability we gain from our probabilistic state representation to *predict* the value $\mathbf{h}_i$ of any measurement and also calculate the uncertainty expected in this measurement in the form of the innovation covariance S_i. Explicitly, our measurement model is:

$$\mathbf{h}_i = \begin{pmatrix} \alpha_{pi} \\ \alpha_{ei} \\ \alpha_{vi} \end{pmatrix} = \begin{pmatrix} \tan^{-1} \frac{h_{Gix}}{h_{Giz}} \\ \tan^{-1} \frac{h_{Giy}}{h_{Gip}} \\ \tan^{-1} \frac{I}{2h_{Gi}} \end{pmatrix},$$

where

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

is the Cartesian vector from the head center to feature i (expressed in the robot-centerd coordinate frame). 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 interocular separation of the active head and H is the height above the ground plane of the head center.

The innovation covariance S_i is calculated as:

$$\mathrm{S}_i = \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v} \mathrm{P}_{xx} \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v}^{\top} + \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v} \mathrm{P}_{xyi} \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i} + \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i} \mathbf{P}_{yix} \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v}^{\top} + \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i} \mathrm{P}_{yiyi} \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i} + \mathrm{R}.$$

Here, R is the measurement noise covariance matrix, defined shortly. Calculating S_i before making measurements allows us to form a search region in measurement space for each feature, at a chosen number of standard deviations (providing automatic gating and minimizing search computation). We will see later that S_i also provides the basis for automatic measurement selection.

In our work, measurement of a feature in the map involves the stereo head (sketched in Fig. 1c) using this prediction to turn to fixate the *expected* position of the feature, carry out a stereo image search of size determined by the innovation covariance (see Section 3.2), and then use its matched image coordinates in combination with the head's known odometry and forward kinematics to produce a measurement $\mathbf{z}_i$ of the position of the feature relative to the robot.

For filtering, measurements are parameterized in terms of the pan, elevation, and (symmetric) vergence angles $\alpha_{p,e,v}$ of an *idealised* active head able to measure the positions of the features at perfect fixation: by idealized, we mean an active head which does not have the small offsets between axes possessed by our head. In image measurements, we expect to detect features to an accuracy of ± 1 pixel, which at the center of the image in the cameras used is an angular uncertainty of about 6×10^{-3} rad. Compared with this, angular errors introduced by the active head, whose axes have repeatabilities two orders of magnitude smaller, are negligible. The advantage of the idealized head parameterization is that when we map the uncertainty coming from image measurements into this space, the measurement

noise covariance is very closely diagonal and constant and can be approximated by:

$$\mathrm{R} = \begin{pmatrix} \Delta\alpha_p^2 & 0 & 0 \\ 0 & \Delta\alpha_e^2 & 0 \\ 0 & 0 & \Delta\alpha_v^2 \end{pmatrix}.$$

In fact, in our system $\Delta\alpha_p = \Delta\alpha_e = \Delta\alpha_v$. This is preferable to parameterizing measurements in the Cartesian space of the relative location of feature and robot since, in that case, the measurement noise covariance would depend on the measurement in a nonlinear fashion (in particular, the uncertainty in depth increases rapidly with feature distance) and this could lead to biased estimates.

2.5 Updating and Maintaining the Map

Once a measurement $\mathbf{z}_i$ of a feature has been returned, the Kalman gain W can then be calculated and the filter update performed in the usual way [20]:

$$\begin{aligned} \mathrm{W} &= \mathrm{P}\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}^{\top} \mathrm{S}^{-1} \\ &= \begin{pmatrix} \mathrm{P}_{xx} \\ \mathrm{P}_{y1x} \\ \mathrm{P}_{y2x} \\ \vdots \end{pmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{x}_v}^{\top} \mathrm{S}^{-1} + \begin{pmatrix} \mathrm{P}_{xyi} \\ \mathrm{P}_{yiyi} \\ \mathrm{P}_{y2y2} \\ \vdots \end{pmatrix} \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i}^{\top} \mathrm{S}^{-1}. \end{aligned}$$

$$\hat{\mathbf{x}}_{new} = \hat{\mathbf{x}}_{old} + \mathrm{W}(\mathbf{z}_i - \mathbf{h}_i)$$

$$\mathrm{P}_{new} = \mathrm{P}_{old} - \mathrm{WSW}^{\top}$$

Since, in our measurement model, the measurement noise R is diagonal, this update can be separated in implementation into separate, sequential updates for each scalar component of the measurement (that is to say that we perform the above update three times, once for each angular component $\alpha_{p,e,v}$ of the measurement; $\mathbf{h}_i$, $\mathbf{z}_i$, and S become scalar in these steps): this is computationally advantageous.

Initializing a New Feature. When an unknown feature n is observed for the first time, a vector measurement $\mathbf{h}_n$ is obtained of its position relative to the head and its state initialized accordingly using the inverse $\mathbf{y}_n(\mathbf{x}_v, \mathbf{h}_n)$ of the measurement model. Jacobians $\frac{\partial \mathbf{y}_n}{\partial \mathbf{x}_v}$ and $\frac{\partial \mathbf{y}_n}{\partial \mathbf{h}_G}$ are calculated and used to update the total state vector and covariance:

$$\mathbf{x}_{new} = \begin{pmatrix} \mathbf{x}_v \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{pmatrix}$$

$$\mathrm{P}_{new} = \begin{pmatrix} \mathrm{P}_{xx} & \mathrm{P}_{xy1} & \cdots & \mathrm{P}_{xx}\frac{\partial \mathbf{y}_n}{\partial \mathbf{x}_v}^{\top} \\ \mathrm{P}_{y1x} & \mathrm{P}_{y1y1} & \cdots & \mathrm{P}_{y1x}\frac{\partial \mathbf{y}_n}{\partial \mathbf{x}_v}^{\top} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \mathbf{y}_n}{\partial \mathbf{x}_v}\mathrm{P}_{xx} & \frac{\partial \mathbf{y}_n}{\partial \mathbf{y}_n}\mathrm{P}_{xy1} & \frac{\partial \mathbf{y}_n}{\partial \mathbf{x}_v}\mathrm{P}_{xy2} & \mathrm{A} \end{pmatrix},$$

where

$$\mathrm{A} = \frac{\partial \mathbf{y}_n}{\partial \mathbf{x}_v}\mathrm{P}_{xx}\frac{\partial \mathbf{y}_n}{\partial \mathbf{x}_v}^{\top} + \frac{\partial \mathbf{y}_n}{\partial \mathbf{h}_n}\mathrm{R}\frac{\partial \mathbf{y}_n}{\partial \mathbf{h}_n}^{\top}.$$

Deleting a Feature. A similar Jacobian calculation shows that deleting a feature from the state vector and covariance matrix is a simple case of excising the rows and columns which contain it. For example, where the second of three known features is deleted, the parts removed are delineated as follows:

$$\begin{pmatrix} \mathbf{x}_v \\ \underline{\mathbf{y}_1} \\ \underline{\mathbf{y}_2} \\ \mathbf{y}_3 \end{pmatrix}, \left(\begin{array}{c|c|c|c} \mathrm{P}_{xx} & \mathrm{P}_{xy1} & \mathrm{P}_{xy2} & \mathrm{P}_{xy3} \\ \hline \underline{\mathrm{P}_{y1x}} & \underline{\mathrm{P}_{y1y1}} & \underline{\mathrm{P}_{y1y2}} & \underline{\mathrm{P}_{y1y3}} \\ \underline{\mathrm{P}_{y2x}} & \underline{\mathrm{P}_{y2y1}} & \underline{\mathrm{P}_{y2y2}} & \underline{\mathrm{P}_{y2y3}} \\ \hline \mathrm{P}_{y3x} & \mathrm{P}_{y3y1} & \mathrm{P}_{y3y2} & \mathrm{P}_{y3y3} \end{array}\right).$$

3 Detection and Matching of Scene Features

Repeatable localization in a particular area requires that reliable, persistent features in the environment can be found and refound over long periods of time. This differs perhaps from the more common use of visual features in structure from motion, where they are often treated as transient entities to be matched over a few frames and then discarded. When the goal of mapping is localization, it is important to remember that motion drift will occur unless reference can be made to features after periods of neglect.

The visual landmarks we will use should be features which are easily distinguishable from their surroundings, robustly associated with the scene geometry, viewpoint invariant, and seldom occluded. In this work, we assume the features to be stationary points.

Since when navigating in unknown areas nothing is known in advance about the scene, we do not attempt to search purposively for features in certain locations which would be good sites for landmarks: There is no guarantee that anything will be visible in these sites which will make a good landmark. Rather, feature acquisition takes place as a data-driven process: The robot points its cameras in rather arbitrary directions and acquires features if regions of image interest are found. This rather rough collection of features is then refined naturally through the map maintenance steps described in Section 5.3 into a landmark set which spans the robot's area of operation.

3.1 Acquiring 3D Features

Features are detected using the Harris corner detector [23] as applied by Shi and Tomasi [24] to relatively large pixel patches (15×15 rather than the usual 5×5 for corner detection). Products of the spatial gradients I_x and I_y of the smoothed image irradiance are averaged over the patch and, if both eigenvalues of the matrix

$$\begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

are large, the patch is corner-like.

To acquire a new feature at the current head position, the detector is run over the left image, finding a predetermined number of the most salient nonoverlapping regions. For the strongest feature, an epipolar line is constructed in the right image (via the known head geometry) and a band around the line searched for a stereo match. If a good 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 robot-centered coordinate frame to be calculated. The head is driven to fixate the feature, enforcing symmetric left and right head

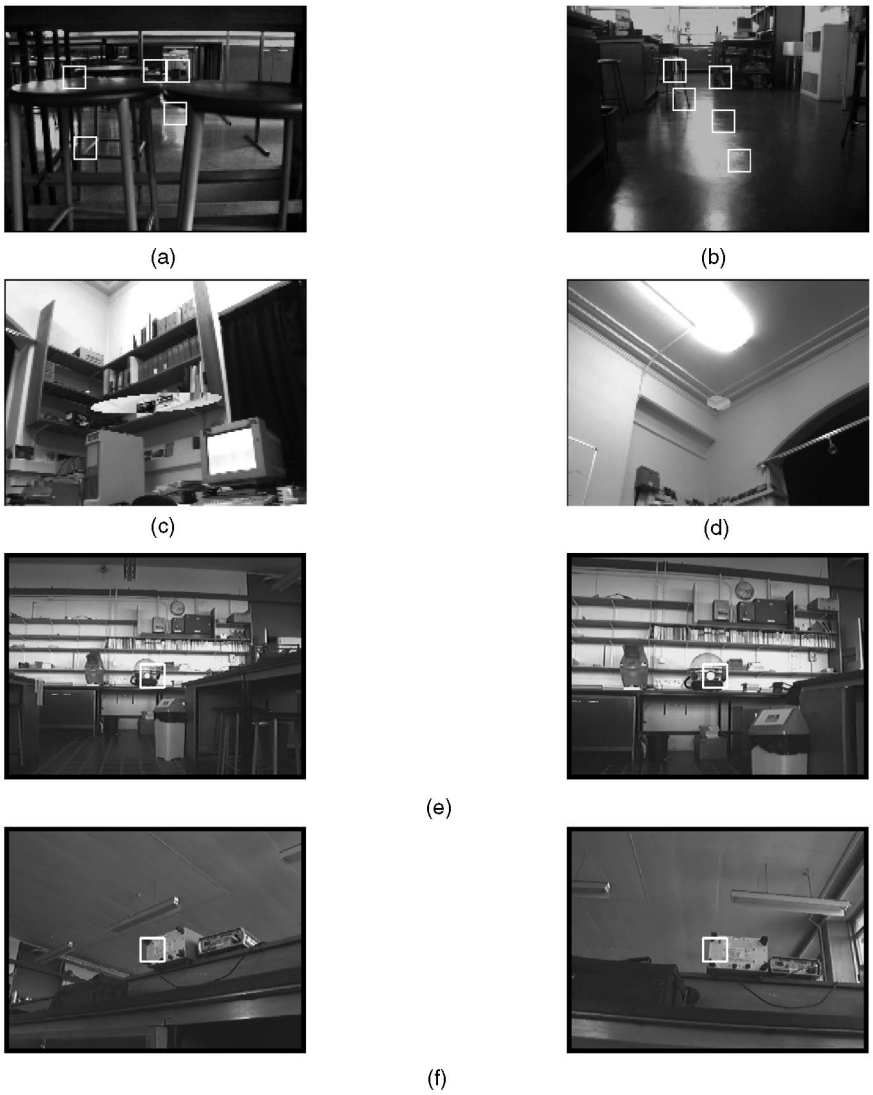

Fig. 2. (a) and (b): Feature detection. Rogue features likely to be deleted as nonstationary arise from depth discontinutites and specularities. (c) and (d): Elliptical search regions generated for features; the size of the ellipse depends on the uncertainty in the relative position of the robot and feature. (e) and (f): Two examples of successful feature matching close to the limits of visibility constraints.

vergence angles to remove redundancy, the feature remeasured, and the process iterated to a given tolerance. Making measurements at fixation reduces dependency on knowledge of the camera focal lengths. The image patch intensity values of the new feature are saved so that appearance matching is possible later and the feature is inserted into the map with uncertainty derived as in Section 2. Note that this uncertainty depends only on the geometrical location of the feature and not on its image characteristics: We assume that image matching (see Section 3.2) has a constant uncertainty in image space; that is to say that how accurately a particular feature can be located in the image does not depend on its appearance.

In our work, as in [24], no attempt is made to discern good or bad features, such as those corresponding to reflections or lying at depth discontinuities (such as those seen in the rather pathological examples of Figs. 2a and 2b) or those which are frequently occluded at the detection stage: The strategy used is to accept or reject features depending on how well they can be tracked once the robot has started moving. Patches which do not actually correspond to stationary, point features will quickly look very different from a new viewpoint, or will not appear in the position expected from the vehicle motion model and, thus, matching will fail (this is also the case with frequently occluded features which are soon hidden behind other objects. These features can then be deleted from the map as will become clearer in our discussion of experiments later: While the initial choice of features is unplanned and random, the best features survive for long periods and become persistent landmarks.

3.2 Searching for and Matching Features

Applying the feature detection algorithm to the entire image is required only to find new features. Since we propagate full information about the uncertainty present in the map, whenever a measurement is required of a particular feature, regions can be generated in the left and right images within which the feature should lie with some desired probability

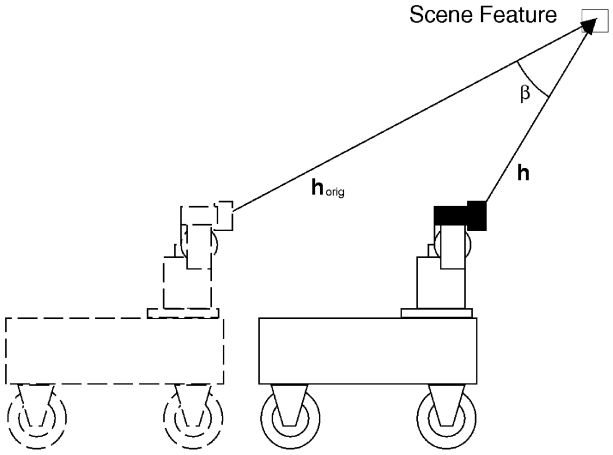

Fig. 3. The expected visibility of a feature is calculated based on the difference in distance and angle between the viewpoint from which it was initially seen and that from which the current measurement is to be made.

(usually three standard deviations from the mean). Typical search ellipses are shown in Figs. 2c and 2d.

Matching within these regions is then achieved by a brute-force correlation search using normalized sum-of-squared-differences for the best match to the saved feature patch within the (usually relatively small) regions defined by the search ellipses in both left and right images. A consistency check is then applied between the two image locations found (taking account of the epipolar coupling between the two measurements): This gives some robustness against mismatches. Normalized sum-of-squared-differences gives the matching a fairly large degree of robustness with respect to changing light conditions and in experiments has meant that the same features could be matched well over the duration of experiments of many minutes or a few hours, though we have not tested the durability of matching under extreme changes (from natural to artificial lighting, for example).

Figs. 2e and 2f show matches obtained for some features, giving an impression of the surprising range of viewpoints which can be matched successfully using the large patch representation of features. However, clearly matching can only be expected to succeed for moderate robot motions since the patch representation is intrinsically viewpoint-variant—features look different when viewed from new distances or angles (to avoid drift, we do *not* update feature templates after matching). Therefore, we have defined a criterion for expected matchability based on the difference between the viewpoint from which the feature was initially seen and a new viewpoint. Fig. 3 shows a simplified cut-through of the situation: $\mathbf{h}_{\text{orig}}$ is the vector from the head center to the feature when it was initialized and $\mathbf{h}$ is that from the head center at a new vehicle position. The feature is expected to be visible if the length ratio $\frac{|\mathbf{h}|}{|\mathbf{h}_{\text{orig}}|}$ is close enough to 1 (in practice between something like $\frac{5}{7}$ and $\frac{7}{5}$) and the angle difference $\beta = \cos^{-1}((\mathbf{h}\cdot\mathbf{h}_{\text{orig}})/(|\mathbf{h}\parallel\mathbf{h}_{\text{orig}}|))$ is close enough to 0 (in practice less than 45° in magnitude); the matches shown in Figs. 2e and 2f are close to these limits of viewpoint change. In our localization algorithm, we are in a position to estimate both of these vectors before a measurement is made and so attempts are made only to measure features which are expected to be visible. The result is a region of the robot's movement space defined for each feature from which it should be able to be seen. A feature which fails to match regularly within this region should be deleted from the map since the failures must be due to it being an essentially "bad" feature in one of the senses discussed above rather than due to simple viewpoint change.

3.3 Failure Modes

Two failure modes were observed in our EKF-based SLAM system. The first arises from failure of data association: Mismatches are likely to happen when robot and feature uncertainty grow and search regions (Figs. 2c and 2d) become very large (for instance, of a width in the region of 100 pixels rather than the more normal 10–20 pixels). In this situation, there is a chance that an image region of similar appearance to a mapped feature is incorrectly identified and this failure cannot be identified by normal measurement gating. In this work, we did not implement a multiple hypothesis framework and, therefore, a single mismatch could prove to be fatal to the localization process. However, mismatches were actually very rare: First, the large size of image patch used to represent a feature meant that matching gave very few false-positives within the uncertainty-bounded search regions (which implicitly impose the explicit consistency checks, based on multifocal tensors, for example, included in most structure from motion systems). More importantly, though, the active measurement selection and map-management approaches used meant that at all times attempts were made to keep uncertainty in the consistency of the map to a minimum. In normal operation, image search regions were small and the chance of mismatches low. For this reason, long periods of error-free localization were possible. Nevertheless, in future systems there is a clear need for an explicit approach to multiple hypotheses.

The second, much rarer, failure mode arose from nonlinearities. When uncertainty in the map is large, measurements with a large innovation may lead to unpredictable EKF updates due to the unmodeled nonlinearity in the system.

4 System Verification Against Ground Truth

To evaluate the localization and map-building accuracy of the system in a controlled environment, the laboratory floor was marked with a grid (to enable manual ground-truth robot position measurements) and artificial scene features were set up in known positions equally spaced in a line along the bench top (Fig. 4a). The robot's motion was controlled interactively in this experiment by a human operator, who also manually indicated (by highlighting image interest regions via the mouse) which features the robot should initialize into its map.

Starting from the grid origin with no prior knowledge of the locations of scene features, the robot was driven nominally straight forward. Every second feature in the line was fixated and tracked for a short while on this outward journey, the robot stopping at frequent intervals so that manual ground-truth measurements could be made of its position and orientation using an onboard laser pointer. The recovered values and uncertainties in the positions of features 0–5 are shown in gray in Fig. 4b, superimposed on the measured ground truth in black. The effects of drift are apparent and the uncertainties have increased steadily.

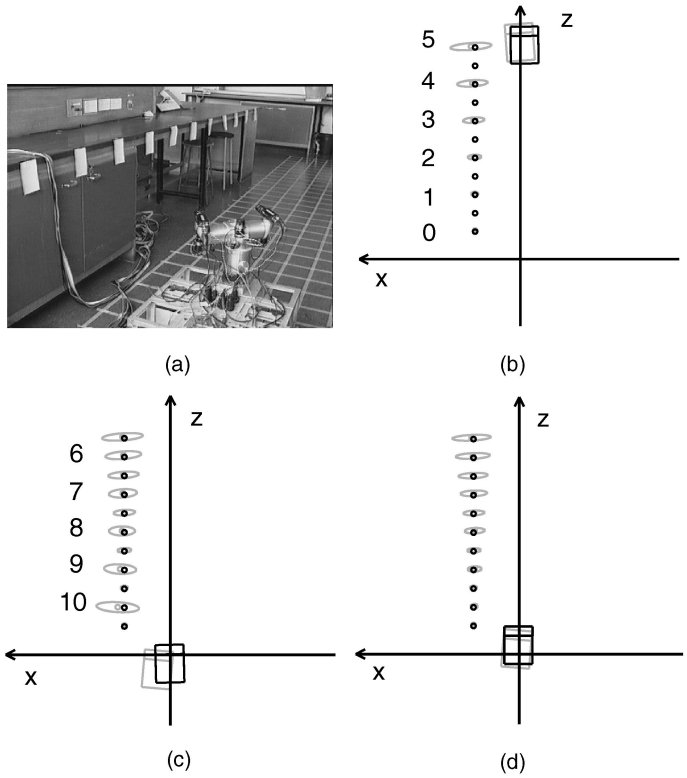

Fig. 4. Experiment with artificially introduced features. Experimental arrangement. Estimated positions of the robot $(\hat{\mathbf{x}}_v)$ and features $(\hat{\mathbf{y}}_i)$ in gray, along with 3σ ellipses for the point covariances $\mathrm{P}_{y_i y_i}$ are shown superimposed on the true positions (from manual measurement) in black as the robot moved forward and back. The feature spacing was 40cm and the robot moved abnout 5m from its origin. Feature labels 0-10 show the order they were tracked in. (As ever with stereo, the major axis of the uncertainty ellipse lies along the Cyclopean direction—and so here the head was viewing on average perpendicular to the direction of travel.) (a) Experimental setup, (b) outward journey, (c) return journey, and (d) feature 0 refound.

The robot was then reversed back down the corridor and made to fixate upon the alternate features it had *not* used previously. The aim was that it should return to its origin while always tracking only recently acquired features, as would be the case in a looped movement around a rectangular layout of corridors, for example. As expected, the uncertainty continued to increase (Fig. 4c) and, by its return to the nominal origin, the filter estimated the robot's position as $z = -0.11$m, $x = 0.26$m, $\phi = -0.08$rad, whereas the true position was $z = 0.01$m, $x = 0.02$m, $\phi = 0.02$rad.

At this stage, following one more movement step, the robot was allowed to refixate feature 0, which it had seen much earlier at the start of the experiment. As can be seen in Fig. 4d, drift and uncertainty are immediately reduced, both in the robot state and scene geometry, particularly near the refixated feature. The estimated position of $z = 0.31$m, $x = 0.04$m, $\phi = -0.08$rad was now much closer to the true position $z = 0.39$m, $x = 0.02$m, $\phi = 0.00$rad. The robot state covariance P_{xx} reduced sharply after refixation from

$$\begin{pmatrix} 0.0039 & -0.0095 & 0.0036 \\ -0.0095 & 0.0461 & -0.0134 \\ 0.0036 & -0.0134 & 0.0051 \end{pmatrix} \rightarrow \begin{pmatrix} 0.0016 & -0.0004 & 0.0016 \\ -0.0004 & 0.0002 & -0.0004 \\ 0.0016 & -0.0004 & 0.0018 \end{pmatrix}.$$

It can be seen that a reasonable degree of uncertainty still remains: this is due to the fact that a single measurement, even of a feature with very small position uncertainty, does not fully constrain the robot's position estimate—further refixations of other features providing complementary information will allow the robot's position to be really locked-down (as will be explained in more detail in Section 5.1).

By maintaining full covariance information, uncertainty grows as a function of actual distance from a known position—here, the origin, where the coordinate frame was defined at the robot's starting position—*not* as a function of the total distance traveled by the robot from the known point. The drift still seen in the uncertainty in distant features is a fundamental limitation of any map-building situation involving the use of sensors with limited range: The locations of these features relative to the world coordinate frame must be estimated by implicit compounding of many noisy measurements and uncertain robot motions.

5 Tools for Autonomous Navigation

The previous experiment was contrived in that the robot was instructed which features to fixate and how to navigate. In this section, we describe tools which combine to permit autonomous active SLAM, as will be demonstrated in the experiments presented later. First, in Sections 5.1 and 5.2, is a method for performing the critical role of actively choosing which feature to fixate upon at each stage of navigation, both without and with consideration of the time penalty involved with refixation using a mechanical device. Next, in Section 5.3, we consider the maintenance of a feature set and, finally, in Section 5.4, discuss how to inject an element of goal-direction into the robot's progress.

5.1 Active Feature Selection

In our SLAM work, the goal is to build a map of features which aids localization rather than an end result in itself. Nevertheless, in the combined and coupled estimation of robot and feature locations which this involves, estimation of the robot position is not intrinsically more "important" than that of the feature positions: aiming to optimize robot position uncertainty through active choices is misleading since it is the overall integrity and consistency of the map and the robot's position within it which is the critical factor. We have already seen in the preceding experiment that robot position uncertainty relative to a world frame will increase with distance traveled from the origin of that frame. It is the mutual, relative uncertainty between features and robot which is key.

Our feature selection strategy achieves this by making a measurement at the currently visible feature in the map whose position is hardest to predict, an idea used in the area of active exploration of surface shape by Whaite and Ferrie [25]. The validity of this principle seems clear: There is little utility in making a measurement whose result is easy to forecast, whereas much is to be gained by making a measurement whose result is uncertain and reveals something new. The principle can also be understood in terms of information theory since a measurement which reduces a widely spread prior probability distribution to a more peaked posterior distribution has a high information content.

Our approach to mapping is active, not in the sense of Whaite and Ferrie who actually control the movement of a camera in order to optimize its utility in sensing surface shape, in that we do not choose to alter the robot's path to improve map estimates. Rather, assuming that the robot trajectory is given or provided by some other goal-driven process, we aim to control the active head's movement and sensing on a short-term tactical basis, making a choice between a selection of currently visible features: Which immediate feature measurement is the best use of the resources available?

To evaluate candidate measurements, we calculate *predicted* measurements $\mathbf{h}$ and innovation covariances S for all visible features (where feature "visibility" is calculated as in Section 3.2). In measurement space, the size of the ellipsoid represented by each S is a normalized measure of the uncertainty in the estimated relative position of the feature and the robot, and we wish to choose the feature with the largest uncertainty. To produce a scalar decision criterion, the volume V_S in $\alpha_{p,e,v}$ space of the ellipsoid at the $n_\sigma = 3\sigma$ level is calculated for each visible feature (an important point here is that, in our implementation, the measurement noise in the three measurement components $\alpha_{p,e,v}$ is a multiple of the identity matrix). Computing the eigenvalues $\lambda_{1,2,3}$ of S yields the volume

$$V_S = (4\pi/3)n_\sigma^3\sqrt{\lambda_1\lambda_2\lambda_3}.$$

We use this measure V_S as our score function for comparing candidate measurements: A measurement with high V_S is hard to predict and, therefore, advantageous to make. Here, we do not propose that V_S is the optimal choice of criterion from an information-theoretic point of view—nevertheless, we believe that it will give results for measurement comparison which are almost identical to an optimal criterion. The important point is that since it is evaluated in a measurement space where the measurement noise is constant, its value reflects how much new information is to be obtained from a measurement and does not a priori favor features which are, for example, close or far from the robot.

An illustrative example is shown in Fig. 5. With the robot at the origin, five well-spaced features were initialized and the robot driven forward and backwards while fixating on feature 0 (chosen arbitrarily). The situation at the end of this motion is shown in Fig. 5a, at which time the five V_S values were evaluated as:

$$V_S(0,1,2,3,4) = (0.04,\ 0.46,\ 1.27,\ 0.49,\ 0.40) \times 10^{-3}.$$

According to our criterion, there is little merit in making another measurement of feature 0 and feature 2 should be fixated instead, rather than 1, 3, or 4. Note here that V_S, being calculated in measurement space, does not necessarily favor those features such as 1 which have large uncertainty in the world coordinate frame. Figs. 5b, 5c, and 5d show the situations which result if features 0, 1, or 2 are fixated for the next measurement. Clearly, making the extra measurement of feature 0 in (b) does little to improve the robot position estimation which has drifted along the direction ambiguous to measurements of that feature. Using features 1 or 2 in Figs. 5c and 5d, however, show significant improvements in robot localization: Visually there is little to choose between these two, but the robot state covariance after fixating feature 2 is smaller:

$$\mathrm{P}_{xx} \times 10^3(\text{if 1 fixated}) = \begin{pmatrix} 0.35 & 0.08 & -0.13 \\ 0.08 & 0.24 & -0.09 \\ -0.13 & -0.09 & 0.10 \end{pmatrix}$$

$$\mathrm{P}_{xx} \times 10^3(\text{if 2 fixated}) = \begin{pmatrix} 0.10 & 0.05 & -0.03 \\ 0.05 & 0.21 & -0.10 \\ -0.03 & -0.10 & 0.09 \end{pmatrix}.$$

The qualities of the V_S above criterion become clear when we consider the case of comparing features *immediately after they have been initialized into the map*; this is a situation we will often face as the robot moves into a new area and stops to find new features. In this case, if the just-initialized features are compared for immediate remeasurement, we find that they all have exactly the same value of V_S:

$$V_{S(\text{new})} = (4\pi/3)(\sqrt{2}n_\sigma)^3\Delta\alpha_p\Delta\alpha_e\Delta\alpha_v.$$

This is an initially surprising but desirable characteristic of V_S: what has happened is that, in initialization, one unit of measurement noise has been injected into the estimate of the position of each feature relative to the robot. When the innovation covariance for remeasurement is calculated, it has a value which is simply this plus one more unit of measurement noise. We have proven that the V_S criterion has no a priori favoritism toward features in certain positions.

To split these identical values, we need to use additional information: in this case, the future heading direction of the robot. We predict the robot's position in a small amount of time and then evaluate V_S for the new features based on this. The result is that we can choose the feature which we expect to give the most information about the robot's future movement. In reality, what happens is that the criterion will

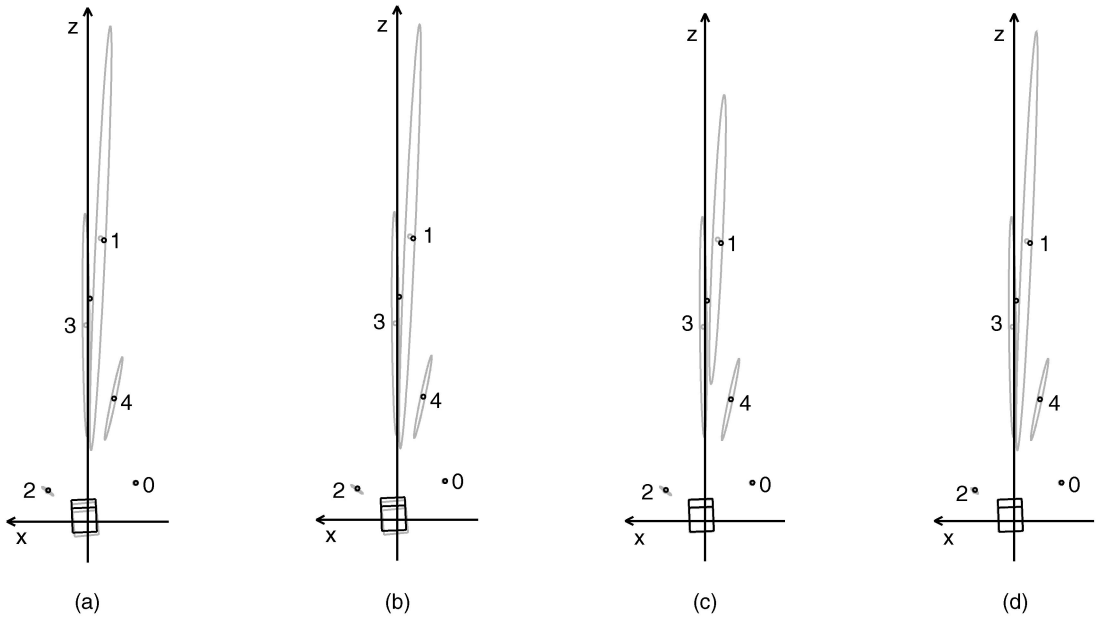

Fig. 5. Selecting between features after a long period tracking one (ground-truth quantities in black, estimates in gray): in (a) the robot stops after tracking feature 0. In (b), (c), and (d), the estimated state is updates after further measurments of features 0, 1, and 2, respectively. The large improvement in the estimated robot state in (c) and (d) shows the value of making measurments of multiple features. (b) Fixate 0, (c) fixate 1, and (d) fixate 2.

choose a feature which will be viewed from a significantly different aspect from the future robot position: When we consider the elongated shape of the measurement noise in our system in Cartesian space, it will choose a feature where from the new position we are able to make a measurement whose covariance ellipse overlaps minimally with the feature's world uncertainty (typically by crossing it at a large angle). This feature provides the best information for reducing future motion uncertainty.

5.2 Measurement Selection during Motion

The strategy developed so far considers measurement choice when the robot is stationary; however, it is not suitable for making active choices actually *while the robot is moving* since it all but demands a change in fixation at every opportunity given to do so. This imperative to switch arises because measuring one point feature does not fully constrain the robot's motion—uncertainty is always growing in one direction or another but predominantly orthogonal to the current fixation direction. This means that switches in fixation are likely to be through around 90° which may take several 100 ms. In fixation switching during motion, we must consider this time delay as a penalty since it could otherwise be spent making different measurements.

We first require a basis for deciding whether one estimated state is better than another. Remembering that total map integrity is what is important, we suggest that the highest V_S found for all visible features, $V_S(\max)$, is a good indicator. If $V_S(\max)$ is high, there is a measurement which needs to be made urgently, indicating that the state estimate is poor. Conversely, if $V_S(\max)$ is low, the relative positions of *all* visible features are known well.

The steps then followed are:

1. Calculate the number of measurements N_i which would be lost during a saccade (a rapid redirection of fixation direction) to each of the visible features. This is done by estimating the time which each head axis would need to move to the correct position, taking the largest (usually the pan time since this axis is the slowest), and dividing by the intermeasurement time interval (200 ms).
2. Identify $N_{\max}$, the highest N_i: This is the number of measurements lost in the largest saccade available.
3. For each feature i, make an estimate of the state after $N_{\max}+1$ steps if an immediate saccade to it is initiated. This consists of making N_i filter prediction steps followed by $N_{\max}-N_i+1$ simulated prediction/measurement updates. A measurement is simulated by updating the state as if the feature had been found in exactly the predicted position (it is the change in covariance which is important here rather than the actual estimate). An estimated state after the same number of steps is also calculated for continued tracking of the currently selected feature.
4. For each of these estimated states, $V_S(\max)$ is evaluated. The saccade providing the lowest $V_S(\max)$ is chosen for action or tracking stays with the current feature if that $V_S(\max)$is lowest.

Fig. 6 shows an experiment into continuous fixation switching: four features were initialized, and Fig. 6a shows the robot's trajectory as it started to move forward, choosing which features to fixate on as described above. In Fig. 6b, the values obtained from a straight V_S comparison of the four features at each time step are plotted. The four lines show how uncertainties in the positions of the features relative to the robot vary with time. As would be hoped, there is a general downward trend from the initial state (where all the features have $V_S = V_{S(\text{new})}$ as explained earlier), showing that the positions are becoming more and more certain.

In the early stages of the motion, fixation switches as rapidly as possible between the four features: Only one measurement at a time is made of each feature before attention is shifted to another. In the graph of Fig. 6b, a measurement of a particular feature appears as a sharp drop in its V_S value. While a feature is being neglected, its V_S gradually creeps up again. This is because the newly-initialized features have large and uncoupled uncertainties: Their relative locations are not well known and measuring one does not do much to improve the estimate of another's position. After a while, the feature states become more coupled: Around step 40, clear zig-zags in the rising curves of neglected features show that the uncertainties in their positions relative to the robot are slightly reduced when a measurement is made of *another* feature.

At around step 80, the first clear situation is seen where it becomes preferable to fixate one feature for an extended period: feature 1 is tracked for about 10 steps. This feature is very close to the robot and the robot is moving towards it: Measurements of it provide the best information on the robot's motion. Since the locations of the other features are becoming better known, their positions relative to the robot are constrained quite well by these repeated measurements (only a gentle rise in the lines for features 0, 2, and 3 is seen during this time). Feature 1 actually goes out of the robot's view at step 101 (the robot having moved too close to it, violating one of the visibility criteria) and behavior returns to quite rapid switching between the other features.

The robot was stopped at the end of this run with state estimates intact. It was then driven back to near the origin in a step-by-step fashion, making further dense measurements of all of the features along the way. The result was that once it was back at its starting point, feature estimates had been very well established. It was from this point that a second continuous switching run was initiated: The trajectory and the now accurately estimated feature positions are shown in Fig. 6c, and a graph of the feature comparison in Fig. 6d.

This second graph is dramatically different from the first: In the early stages, low V_S values for all the features are now maintained by extended periods of tracking one feature (feature 1 again). The strong coupling now established between feature estimates means that if the robot position relative to one can be well estimated, as is the case when the nicely placed feature 1 is tracked, its position relative to the others will be as well. There is the occasional jump to another feature, appearing as spikes in the traces at around steps 70 and 90. Just after step 120, feature 1 goes out of view and a period of rapid switching occurs. None of the remaining features on its own provides especially good overall robot position information and it is necessary to measure them in turn.

Feature 0 goes out of view (due to too large a change in viewing angle) at step 147. After this, only the distant features 2 and 3 remain for measurements. It is noticeable that throughout the graph these two have been locked together in their V_S values: Measurements of them provide very similar information due to their proximity and there is little need to switch attention between them. These features finally go out of view at about step 270, leaving the robot to navigate with odometry only.

A further experiment was performed to investigate the effect of using a head with a lower performance. Software velocity limits were introduced, increasing the head's time to complete saccades by some 30 percent. Runs were made with both fast and slow performances. Two distant features (features 2 and 3 in the previous experiment) were initialized from the origin and the robot drove straight forward, switching attention between them. The results were as one would anticipate. The fast head was able to keep the errors on both points of similar size and continued to switch fixation at a constant rate throughout the run. The slow head was less able to keep the error ratio constant and, later in the run when the feature estimates were well coupled, the rate of switching fell. The larger penalty of slower saccades meant that it was worthwhile tracking one feature for longer.

5.3 Automatic Map Growing and Pruning

Our map-maintenance criterion aims to keep the number of reliable features visible from any robot location close to a value determined by the specifics of robot and sensor, the required localization accuracy, and the computing power available: In this work, the value two was chosen because measurements of two widely-spaced features are enough to produce a fully-constrained robot position estimated.

Features are added to the map if the number visible in the area the robot is passing through is less than this threshold: The robot stops to detect and initialize new features in arbitrarily chosen, widely-spaced viewing directions. This criterion was imposed with efficiency in mind—it is not desirable to increase the number of features and add to the computational complexity of filtering without good reason—and the gain in localization accuracy from adding more features than this minimum would not be great. However, in future work, it may be useful to ensure that one or two features more than the minimum are always visible to ensure that adding new features does not happen too late and the robot is not ever left in a position with less than the minimum available.

A feature is deleted from the map if, after a predetermined number of detection and matching attempts when the feature should be visible, more than a fixed proportion (in our work 50 percent) are failures. This is the criterion which prunes the "bad" features discussed in Section 3.1. In our current implementation, there is no rule in place to ensure that the scene objects corresponding previously deleted features (which are of interest to the feature detection algorithm despite their unsuitability as long-term landmarks) are not acquired again in the future but, in practice, this was rare due to the fact that the robot rarely passes along exactly the same route twice.

It should be noted that a degree of clutter in the scene can be dealt with even if it sometimes occludes landmarks. As long as clutter does not too closely resemble a particular landmark and does not occlude it too often from viewing positions within the landmark's region of expected visibility, attempted measurements while the landmark is occluded will simply fail and not lead to a filter update. The same can be said for moving clutter, such as people moving around the robot, who sometimes occlude landmarks—a few missed measurements are not a big issue. Problems only arise if mismatches occur due to a similarity in appearance between clutter and landmarks and this can potentially lead to catastrophic failure. The correct operation of the system relies on the fact that in most scenes, very similar objects do not commonly appear in a close enough vicinity to lie within a single image search region (and special steps would need to

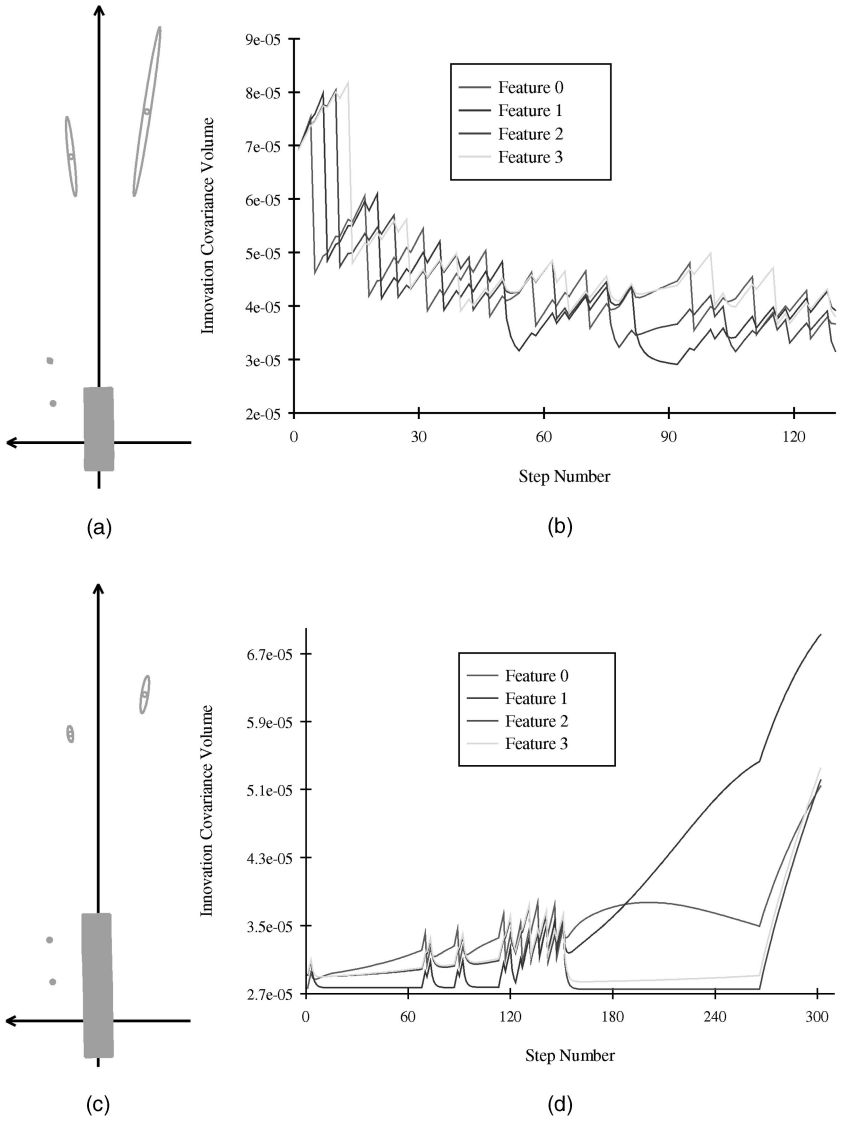

Fig. 6. The innovation covariance volume V_S values and fixation switching (b) as the robot moves forward in the region of four newly-initalized features shown in (a). Each line, representing one feature, drops sharply as that feature is measured and its uncertainty decreases. Later, in the run (e.g., near step 90), extended fixation on one feature becomes preferable to rapid switching. A general downward trend shows continuous improvement in estimates. Parts (c) and (d) show the same for a longer second run, where the geometry is better known from the start. Now, low V_S values for all features are maintained predominantly by long periods tracking one feature. Changes in behavior are seen when feature 1 goes out of view at step 120, feature 0 at step 147 and, finally, features 2 and 3 at step 270, after which all V_S values grow without bound.

be taken to enable the system to work in scenes with a lot of repeated texture).

5.4 Goal-Directed Navigation

The purpose of this paper is to build a map which aids localization rather than one dense enough to be useful for identifying free space. Nevertheless, this localization method could form part of a complete system where an additional module (visual or otherwise) could perform this role and communicate with the localization system to label some of its features with contextual information, such as "this is a feature at the left-hand side of an obstacle."

In an earlier paper [26], we showed how fixation could be used to steer a vehicle toward and then around a fixated waypoint and then on to the next waypoint. The method produces steering outputs similar to those of human drivers [27]. In Fig. 7, we show an image sequence obtained from one of the robot's cameras in a period of fixation tracking of a certain map feature and the path followed by the robot during such a maneuver. Section 8 shows how this type of behavior can be incorporated into the mapping system.

6 Automatic Position-Based Navigation

With automatic feature-selection, map maintenance, and goal-directed steering, the robot is in a position to perform autonomous position-based navigation. A trajectory is specified as a sequence of waypoints in the world coordinate frame through which the robot is desired to pass. The robot moves in steps of approximately two seconds duration. Before each step, the feature selection algorithm of the previous section chooses the best feature to track during the movement and this feature is tracked continuously during movement (at a rate of 5Hz, making 10

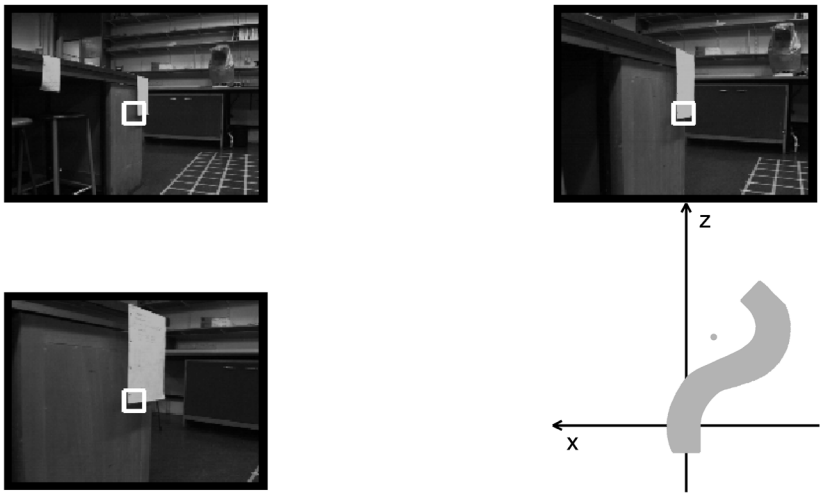

Fig. 7. Image sequence obtained from continuous fixation tracking of a feature while following an avoidance path generated by a biologically-inspired control law.

measurements and the same number of filter prediction/update steps per movement step). The robot stops for a short period between movement steps to make a gross fixation change to another feature. The breaks in movement are also used to automatically add features to or delete them from the map as necessary. As the robot drives making measurements of the chosen feature and updating the localization filter, the steering angle is continuously set to the appropriate value to reach the next waypoint.

In the follow experiment, 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, back to $(0, 0)$ again (in meter units). This experiment was designed to prove again the system's ability to return to a previously visited area and recognize it as such, but now using a map which was generated and maintained completely automatically. (The extra waypoint $(6, 0.4)$ was specified merely to ensure that the robot turned in a way which did not snag its umbilical cable.)

The robot's progress is shown in Fig. 8, along with views from the left camera of some of the first 15 features inserted into the map, which itself is shown at various stages in Fig. 9.

On the outward journey, the sequence of features fixated in the early stages of the run (up to step (21)) 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. Features 4 and 5 did not survive past very early measurement attempts and do not appear in Fig. 9. 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 on the furthest wall (11–15, etc.). The active approach really comes into its own during sharp turns such as that made around step (44), where using the full range of the pan axis features such as these could be fixated while the robot made a turn of $180°$. The angle of turn can be estimated accurately at a time when wheel odometry data is particularly unreliable.

At step (77), the robot had reached the final waypoint and returned to its starting point. The robot successfully rematched original features on its return journey, in particular feature 0.

The robot's true position on the grid compared with the estimated position was ($\mathbf{x}_v = (z, x, \phi)^\top$ being given in meter and radian units):

$$\mathbf{x}_v = (0.06,\ -0.12,\ 3.05)^\top, \quad \hat{\mathbf{x}}_v = (0.15,\ -0.03,\ 2.99)^\top.$$

To verify the usefulness of the map generated, the experiment was continued by commanding the robot to repeat the round trip. In these further runs, the system needed to do little map maintenance—of course, all measurements add to the accuracy of the map but there was little need to add to or delete from the set of features stored because the existing set covered the area to be traversed well. At (6, 0), step (124), the veridical and estimated positions were

$$\mathbf{x}_v = (5.68,\ 0.12,\ 0.02)^\top,\ \hat{\mathbf{x}}_v = (5.83,\ 0.12,\ 0.02)^\top$$

and on return to the origin, after a total trip of 24m,

$$\mathbf{x}_v = (0.17,\ -0.07,\ -3.03)^\top, \hat{\mathbf{x}}_v = (0.18,\ 0.00,\ -3.06)^\top.$$

A pleasing aspect of the feature choice criterion described earlier is its inbuilt pressure to create tightly known and globally consistent maps. Because uncertainty in the robot's position relative to earlier-seen features expands during the period of neglect, the criterion makes them prime candidates for fixation as soon as they become visible again; reregistration with the original world coordinate frame, in which the locations of these early features is well-known, happens as a matter of course.

7 Incorporating Sparse Prior Knowledge

The fundamental limitation of SLAM is that as the robot moves further from its fudicial starting point, position estimates relative to the world frame become increasingly uncertain and this can be mitigated in many real application domains if there are some visual landmarks which are in positions known in advance. Ideally, they

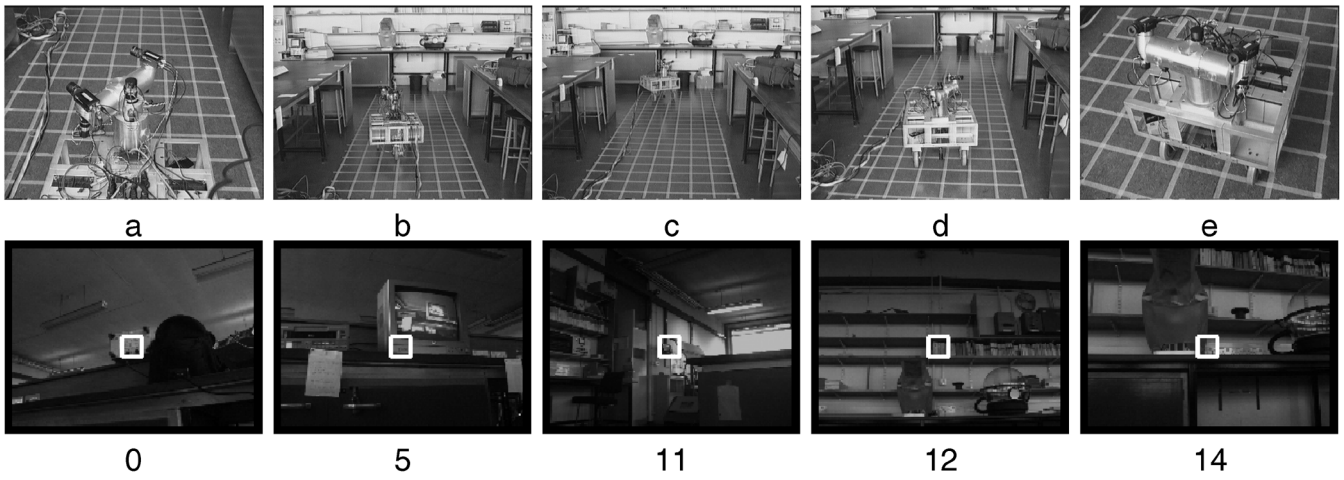

Fig. 8. Frames from a video of the robot navigating autonomously up and down the corridor where the active head can be seen fixating on various features and fixated views from one of its cameras of some of the first 15 features initalized. The gridded floor was an aid to manual ground-truth measurements and was not used by the vision system.

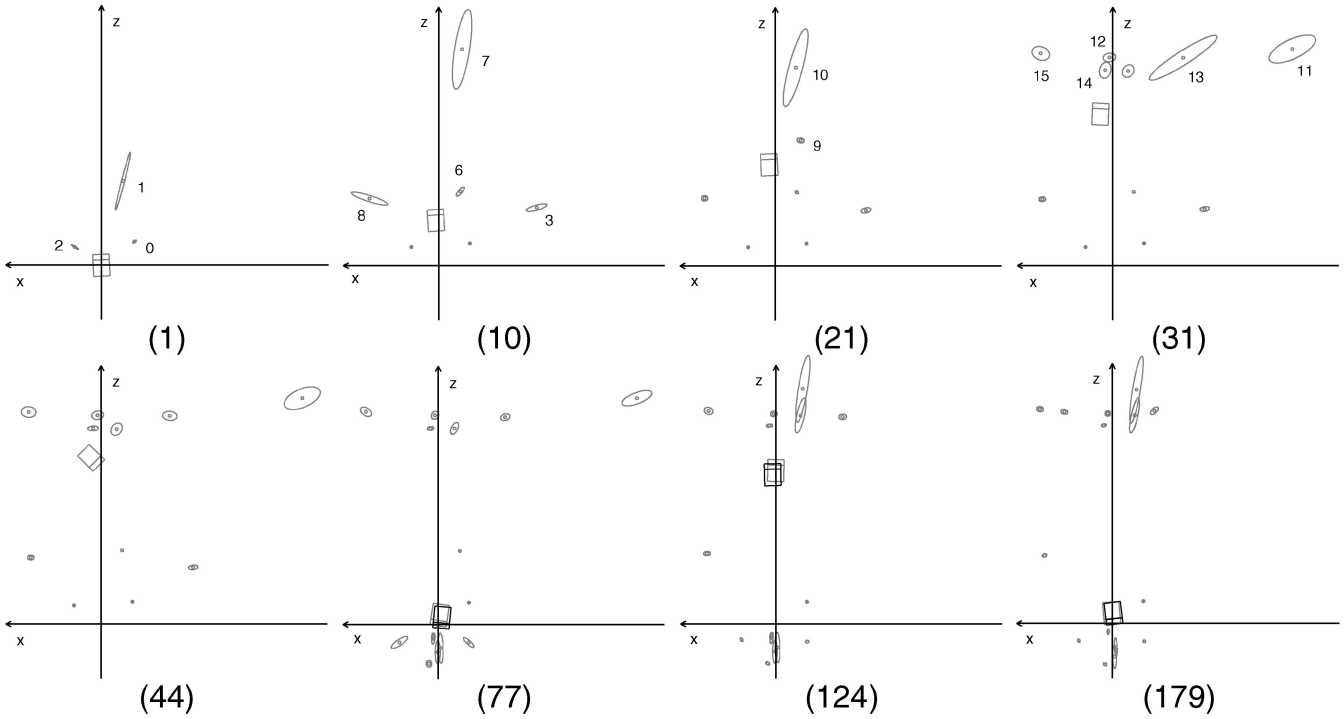

Fig. 9. Numbered steps in autonomous navigation up and down a corridor. Gray shows the estimated locations of the robot features and black (where measured) the true robot position. The furthest features lie at $z \approx 8\mathrm{m}$.

would be distributed uniformly around the mapped area. They must also be visually distinguishable from other features which could, within the growing uncertainty bounds, be mistaken for them: however, this can be more easily achieved with these hand-picked features then those detected autonomously by the robot. There have been many approaches to robot localization using landmarks in known locations: when a map is given in advance, the localization problem becomes relatively simple [4]. Here, however, we wish to show that a small number of natural visual landmarks (small in the sense that there are not enough to permit good localization using only these landmarks) can be easily integrated into the SLAM framework to improve localization.

The landmark's known location is initialized into the estimated state vector as the coordinates $\mathbf{y}_i$ of a feature i at the start of the run (i.e., as though it had already been observed) and its covariance $\mathrm{P}_{y_i y_i}$ is set with all elements equal to zero, along with the cross-covariances between the feature state and that of the robot and other features. In prediction and measurement updates, the filter handles these perfectly known landmarks just like any other feature. Note, however, that uncertainty in a landmark's relative position will grow as the robot moves before observing it, and so the V_S criterion will, as ever, make the landmark desirable to look at.

When there are perfectly known features in the map, it is these which define the world coordinate frame, rather than

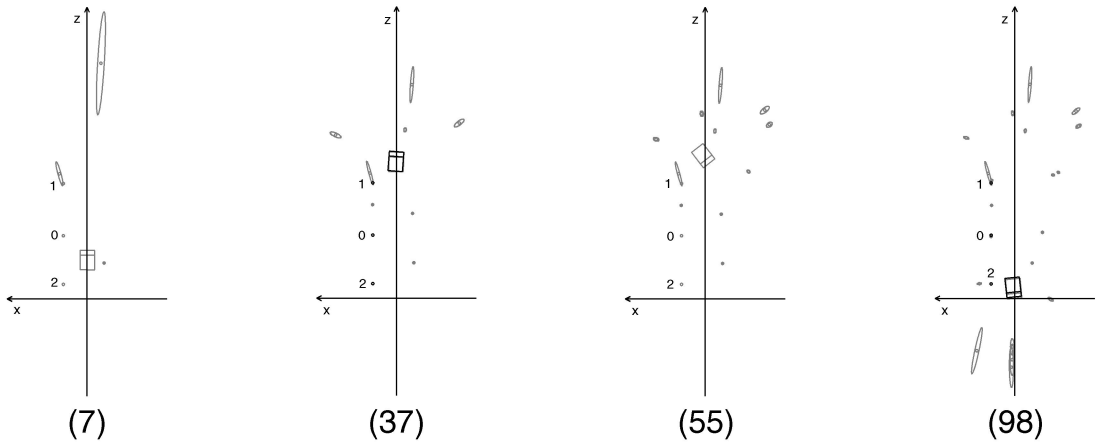

Fig. 10. Automatic position-base navigation with three known features (0, 1, and 2). High localization accuracy can now be achieved over a wider range of robot movement.

the arbitrary definition of this frame at the robot's starting position used before. Therefore, in this experiment, the robot's position was initialized with a starting uncertainty not equal to zero: An assessment was made of the uncertainty in robot location and orientation relative to the known landmarks (with standard deviation of the order of a few centimetres and degrees) and this formed the initial P_{xx}. Note too that as well as perfectly known landmarks, it would be straightforward to introduce landmarks in partially known positions (i.e., with some uncertainty) into this framework.

An experiment was conducted where the robot made a movement similar to that in the autonomous navigation experiment presented earlier, but now with three known features inserted the map before it set out. These lay to one side of the corridor and are labeled as 0, 1, and 2 in the pictures of Fig. 10 showing the progress of the experiment. In just the same way that in the previous experiment the automatic feature-choice criterion selected features not measured for a long time whenever possible, in this experiment the known features were selected as soon as they became visible, showing the drift which was occurring in the robot's estimation relative to the world frame. The benefit of the known features was to improve world-frame localization accuracy when the robot was a long way from its origin. At step (37), when the robot was at its farthest distance from the origin, its ground-truth location was measured. The true and estimated locations were

$$\mathbf{x}_v = (5.83,\ 0.01,\ -0.01)^\top,\ \hat{\mathbf{x}}_v = (5.81,\ 0.01,\ -0.02)^\top,$$

and the covariance matrix an order of magnitude smaller than that achieved earlier.

It can also be seen that the "natural" features initialized close to the landmark are now more certain: The features at the far end of the corridor (high z) in Fig. 10 have much smaller ellipses than those in Fig. 9.

A lateral slice through 3D map recovered in this experiment (Fig. 11a) reveals a curiosity—the use of a virtual reflected feature. The experiment was carried out at night under artificial lighting and as the robot returned to its starting position, it inserted the reflection of one of the ceiling lights into the map as feature 32.

8 Adding Context to a Map

Well-located visual landmarks spread through the scene allow the robot to remain true to the world coordinate frame over a wider area, making navigation by specifying waypoints viable. But, it is also likely that features, whether those supplied to the robot manually or detected automatically, also have contextual meaning and can have labels attached such as "feature 0 is a point on the edge of an obstacle region" or "... is the door jamb." This information could be attached by a human operator or supplied by another visual process.

To illustrate the use of all the techniques developed in this paper for autonomous localization and navigation while map-building, the locations of just two landmarks at the corners of a zig-zag path were given to the robot, 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 using the following plan:

Landmark 0 is Obstacle A at $(z, x) = (5.50, -0.50)$
Landmark 1 is Obstacle B at $(z, x) = (7.60, -2.15)$

1. Go forward to waypoint $(z, x) = (2.0, 0.0)$.
2. Steer round Obstacle A, keeping to the left.
3. Steer round Obstacle B, keeping to the right.
4. Go forward to waypoint $(z, x) = (8.5, -3.5)$.
5. Stop.

In this experiment, steering around the known obstacles took place on a positional basis—the robot steered so as to avoid the known obstacles based on its current position estimate, even before it had first measured them. The automatic feature-selection criterion decided when it was necessary actually to measure the known features and, in the experiments, this proved to be as soon as they became visible in order to lock the robot position estimate down to the world frame. The results are shown in Fig. 12, where the estimated trajectory generated is pictured next to stills from a video of the robot.

The point when a first measurement of known feature 0 is made can be clearly seen in Fig. 12 as a small kink in the robot trajectory: Measuring the feature corrected the robot's drifting position estimate and meant that the steering angle was changed slightly to correct the approach. After this, the obstacle feature was only fixated on when it again became the

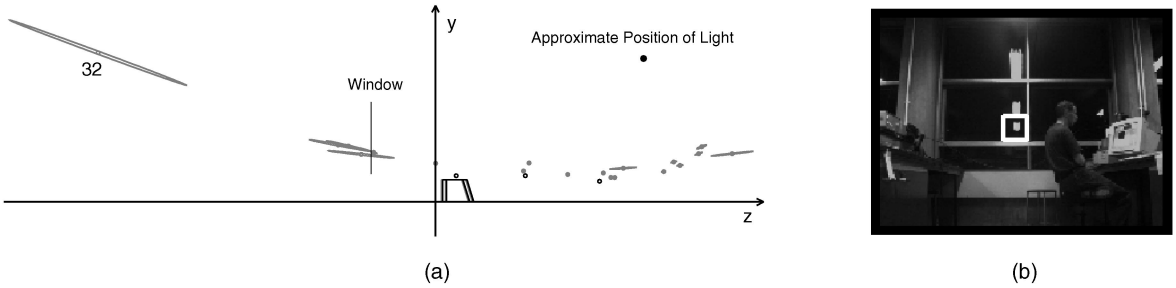

Fig. 11. A virtual reflected feature: 32 is a reflection in a window of an overhead light. Its position in the map lies outside of the laboratory, but it still acts as a stable landmark.

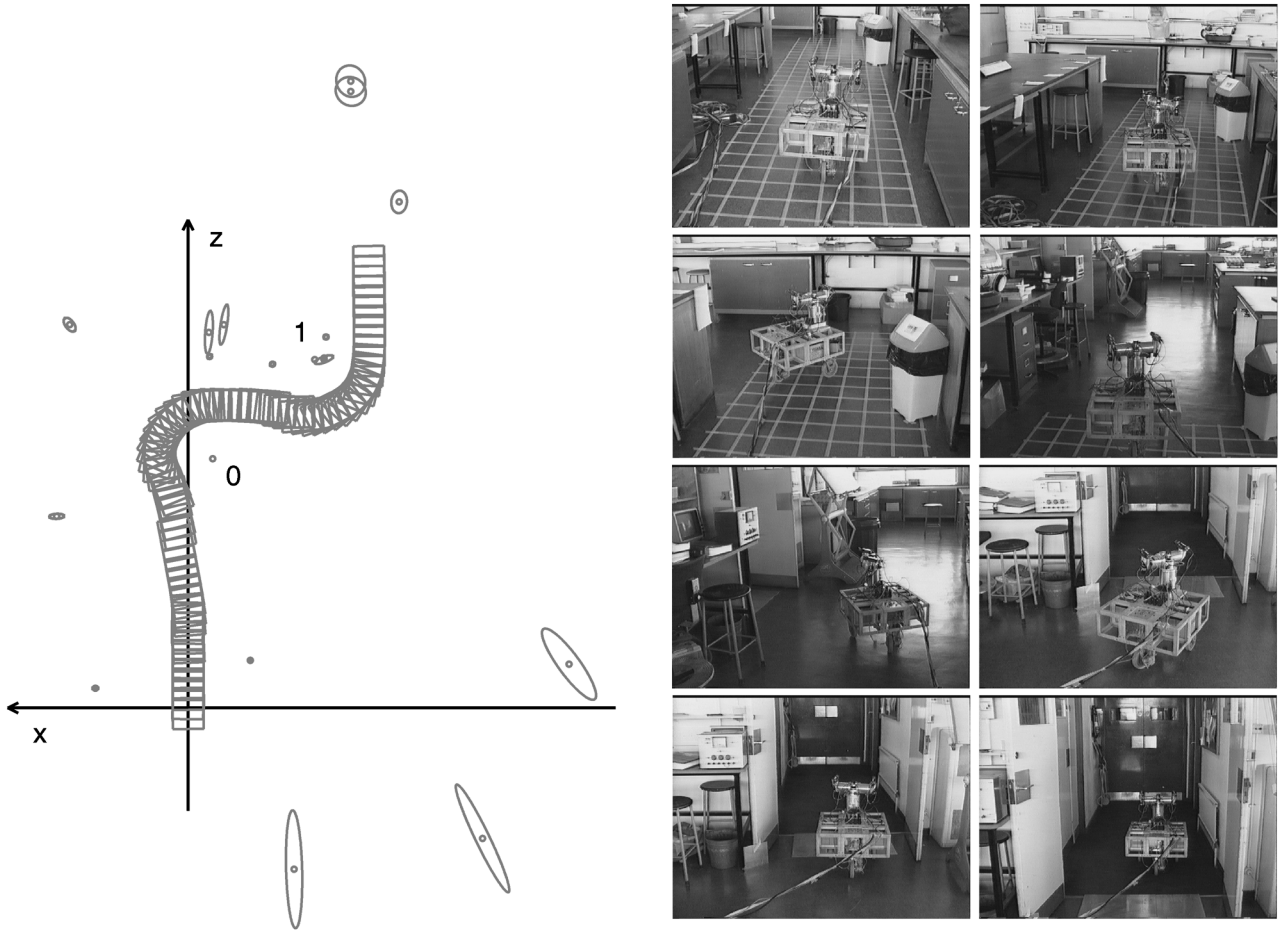

Fig. 12. The estimated trajetory and frames cut from a video as the robot navigated autonomously around two known landmarks and out of the laboratory door. The robot knew the locations of features 0 and 1 as prior knowledge, along with information on their status as obstacles.

best measurement to make. Otherwise, attention was paid to improving the map of automatically-acquired features.

9 Conclusions

We have shown that an active approach is the device which permits vision to be used effectively in simultaneous localization and map-building for mobile robots and presented a fully autonomous real-time implementation.

Here, our use of active vision for navigation differs fundamentally from that explored by Sandini and Tistarelli [28], [29], [30] whose emphasis was on an active approach to recovering free space by computing time to contact from the evolution of disparity and motion parallax. Their representation was dense rather than sparse. The approach here also differs from our earlier work where we utilized an active head for navigation tasks, such as steering around corners and along winding roads [26]. Our results indicate that active fixation has a part to play not only in short-term or tactical navigation tasks, but also in strategic tasks where *informed* visual search is required.

From this position, visual navigation research can join with that progressing using other sensor types and move toward solving the remaining problems in the burgeoning field of sequential map-building. It is also hoped that by introducing the problems of robot map-building to researchers in visual reconstruction, insights can be gained

into the methodology which will be needed to construct structure from motion systems which can operate in real time, the first examples [31] of which have just started to appear.

ACKNOWLEDGMENTS

MPEG video illustrating aspects of this work is available at http://www.robots.ox.ac.uk/ActiveVision/Research/gti.html. The Scene Library, open-source C++ software for simultaneous localization and map-building which evolved from the work described in this paper, is available at http://www.robots.ox.ac.uk/~ajd/Scene/.

REFERENCES

[1] R. Smith, M. Self, and P. Cheeseman, "A Stochastic Map for Uncertain Spatial Relationships," *Proc. Fourth Int'l Symp. Robotics Research,* 1987.
[2] C.G. Harris and J.M. Pike, "3D Positional Integration from Image Sequences," *Proc. Third Alvey Vision Conf.,* pp. 233–236, 1987.
[3] N. Ayache, *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception,* Cambridge, Mass: MIT Press, 1991.
[4] H. F. Durrant-Whyte, "Where am I? A Tutorial on Mobile Vehicle Localization," *Industrial Robot,* vol. 21, no. 2, pp. 11–16, 1994.
[5] C.G. Harris, "Geometry from Visual Motion," *Active Vision,* A. Blake and A. Yuille, eds., 1992.
[6] P.A. Beardsley, I.D. Reid, A. Zisserman, and D.W. Murray, "Active Visual Navigation Using Non-Metric Structure," *Proc. Fifth Int'l Conf. Computer Vision,* pp. 58–65, 1995.
[7] J.-Y. Bouget and P. Perona, "Visual Navigation Using a Single Camera," *ICCV5,* pp. 645–652, 1995.
[8] M. Pollefeys, R. Koch, and L. Van Gool, "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters," *Proc. Sixth Int'l Conf. Computer Vision,* pp. 90–96, 1998.
[9] P.H.S. Torr, A.W. Fitzgibbon, and A. Zisserman, "Maintaining Multiple Motion Model Hypotheses over Many Views to Recover Matching and Structure," *Proc. Sixth Int'l Conf. Computer Vision,* pp. 485–491, 1998.
[10] H.F. Durrant-Whyte, M.W.M. G. Dissanayake, and P.W. Gibbens, "Toward Deployments of Large Scale Simultaneous Localization and Map Building (SLAM) Systems," *Proc. Ninth Int'l Symp. Robotics Research,* pp. 121–127, 1999.
[11] K.S. Chong and L. Kleeman, "Feature-Based Mapping in Real, Large Scale Environments Using an Ultrasonic Array," *Int'l J. Robotics Research,* vol. 18, no. 2, pp. 3–19, Jan. 1999.
[12] S. Thrun, D. Fox, and W. Burgard, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Machine Learning,* vol. 31, 1998.
[13] J.A. Castellanos, "Mobile Robot Localization and Map Building: A Multisensor Fusion Approach," PhD thesis, Universidad de Zaragoza, Spain, 1998.
[14] J.J. Leonard and H.J.S. Feder, "A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization," *Robotics Research,* Springer Verlag, 2000.
[15] A.J. Davison and D.W. Murray, "Mobile Robot Localization Using Active Vision," *Proc. Fifth European Conf. Computer Vision,* pp. 809–825, 1998.
[16] S.K. Nayar, "Catadioptric Omnidirectional Camera," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1997.
[17] A.J. Davison and N. Kita, "Active Visual Localization for Cooperating Inspection Robots," *Proc. IEEE/RSJ Conf. Intelligent Robots and Systems,* 2000.
[18] J.G.H. Knight, A.J. Davison, and I.D. Reid, "Constant Time SLAM Using Postponement," *Proc. IEEE/RSJ Conf. Intelligent Robots and Systems,* 2001.
[19] A.J. Davison, "Mobile Robot Navigation Using Active Vision," PhD thesis, Univ. of Oxford, available at http://www.robots.ox.ac.uk/~ajd/, 1998.
[20] A.J. Davison and N. Kita, "Sequential Localization and Map-Building for Real-Time Computer Vision and Robotics," *Robotics and Autonomous Systems,* vol. 36, no. 4, pp. 171–183, 2001.
[21] J. MacCormick and M. Isard, "Partitioned Sampling, Articulated Objects and Interface-Quality Hand Tracking," *Proc. Sixth European Conf. Computer Vision,* 2000.
[22] S. Thrun, W. Burgard, and D. Fox, "A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping," *Proc. IEEE Int'l Conf. Robotics and Automation,* 2000.
[23] C.G. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Fourth Alvey Vision Conf.,* pp. 147–151, 1988.
[24] J. Shi and C. Tomasi, "Good Features to Track," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 593–600, 1994.
[25] P. Whaite and F.P. Ferrie, "Autonomous Exploration: Driven by Uncertainty," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 3, pp. 193–205, 1997.
[26] D.W. Murray, I.D. Reid, and A.J. Davison, "Steering without Representation with the Use of Active Fixation," *Perception,* vol. 26, pp. 1519–1528, 1997.
[27] M.F. Land and D.N. Lee, "Where We Look When We Steer," *Nature,* vol. 369, pp. 742–744, 1994.
[28] G. Sandini and M. Tistarelli, "Robust Obstacle Detection Using Optical Flow," *Proc. IEEE Int'l Workshop Robust Computer Vision,* 1990.
[29] M. Tistarelli and G. Sandini, "Dynamic Aspects in Active Vision," *Proc. CVGIP: Image Understanding,* vol. 56, no. 1, pp. 108–129, 1992.
[30] E. Grossi and M. Tistarelli, "Active/Dynamic Stereo Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 11, pp. 1117–1128, 1995.
[31] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "'MFm': 3-D Motion from 2-D Motion Causally Integrated over Time," *Proc. Sixth European Conf. Computer Vision,* 2000.

Andrew J. Davison received the BA degree (first-class honors) in physics from the University of Oxford in 1994. His doctoral research, of which the work in this paper formed a part, was in the area of robot navigation using active vision. On completing the DPhil degree in early 1998, he was awarded a European Union Science and Technology Fellowship and worked for two years at the Japanese Government's Electrotechnical Laboratory in Tsukuba, Japan, continuing research into visual navigation for single and multiple robots. He returned to the UK in 2000 and is currently once again working at the University of Oxford, conducting research into real-time localization for arbitrary camera-based robots and devices and visual tracking of humans.

David W. Murray received the BA degree (first-class honors) in physics from the University of Oxford in 1977 and continued there to complete a DPhil in low-energy nuclear physics in 1980. He was research fellow at the California Institute of Technology before joining the General Electric Company's research laboratories in London, where his primary research interests were in motion computation and structure from motion. He moved to the University of Oxford in 1989, where he is now a professor of Engineering Science and a Drapers' fellow in Robotics at St. Anne's College. His research interests have centered for some years on applying model-based vision and stereo-motion computation to active and telerobotic systems. He has published more than 100 articles in journals and refereed conferences in vision and physics, and is coauthor of *Experiments in the Machine Interpretation of Visual Motion* (MIT Press, 1990). He is a fellow of the Institution of Electrical Engineers in the UK. He is a member of the IEEE and the IEEE Computer Society.