Real-Time Localisation and Mapping with Wearable Active Vision [*]

Andrew J. Davison, Walterio W. Mayol and David W. Murray
Robotics Research Group
Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK
http://www.robots.ox.ac.uk/ActiveVision/

Abstract

We present a general method for real-time, vision-only single-camera simultaneous localisation and mapping (SLAM) — an algorithm which is applicable to the localisation of any camera moving through a scene — and study its application to the localisation of a wearable robot with active vision. Starting from very sparse initial scene knowledge, a map of natural point features spanning a section of a room is generated on-the-fly as the motion of the camera is simultaneously estimated in full 3D. Naturally this permits the annotation of the scene with rigidly-registered graphics, but further it permits automatic control of the robot's active camera: for instance, fixation on a particular object can be maintained during extended periods of arbitrary user motion, then shifted at will to another object which has potentially been out of the field of view. This kind of functionality is the key to the understanding or "management" of a workspace which the robot needs to have in order to assist its wearer usefully in tasks. We believe that the techniques and technology developed are of particular immediate value in scenarios of remote collaboration, where a remote expert is able to annotate, through the robot, the environment the wearer is working in.

1 Introduction

A body-mounted visual sensor provides a wearable computer with the opportunity to sense the world from a first-person perspective. The sensor moves with and observes the places attended by the wearer — not only enhancing the chances of recovering the wearer's state from a privileged position but providing non-invasive sensing of the surroundings which reduces the burdening of the environment with technological creepers. When the wearable sensor is under its own control, such that its viewpoint can be moved and thus a certain amount of independence gained from the wearer, the system is better described by the term "wearable visual robot".

A wearable visual robot is in a position to detect both the actions of its wearer (e.g. grasping motions) and the state of the environment (e.g. the pot is boiling), and therefore has the potential to act as an assistant to a wearer working in various domains. It could act as an aid to memory in a construction scenario, helping the wearer keep track of tools and materials; it could provide warnings of dangerous situations or objects, or draw attention to those of interest; with possession of specific domain knowledge it could provide the wearer with a work-plan — perhaps guiding the medical treatment administered by a non-expert paramedic in a remote location.

A more immediate mode of operation than fully autonomous assistant would be to envisage the wearable robot as the facilitator of help from a remote human expert, as depicted in Figure 1. In such remote collaboration, augmenting and mediating the environment with information is attractive if information can be positioned relative to specific objects or places, achieving a true context-specific flow of information. This augmentation can be beneficial to both the wearer and to the remote expert. The remote expert could for example drop virtual notes on top of the objects present in the space browsed by the wearer. These notes could be navigation clues, warnings, things to do or any other sort of information that could be of use to the current expert/wearer partnership or to other future users.

1.1 Wearable Vision and Localisation

If a wearable robot equipped with a vision system is to assist its user, either autonomously or via remote annotation, the robot must know where it is with respect to objects of interest in the surroundings, whether these objects are known in advance or detected autonomously. While various other sensor types (often requiring additional scene infrastructure, such as magnetic or ultrasonic sensors) can pro-

[*] This research is supported by an EPSRC Advanced Research Fellowship to AJD, UK EPSRC Grant GR/N03266 and a Mexican CONACYT scholarship to WWM. The authors are particularly grateful to B.J. Tordoff for software collaboration.

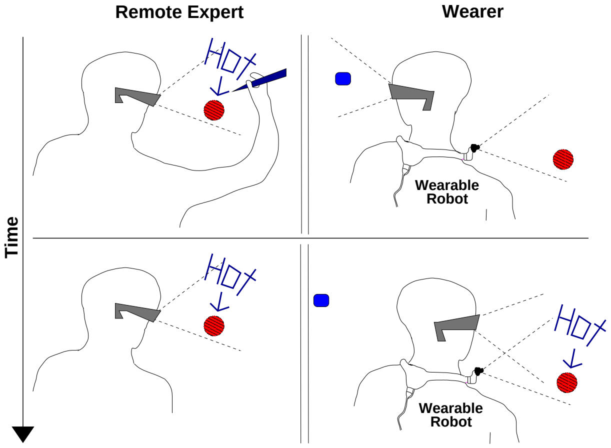


Figure 1. A remote expert annotates the environment he interacts with via a wearable visual robot.

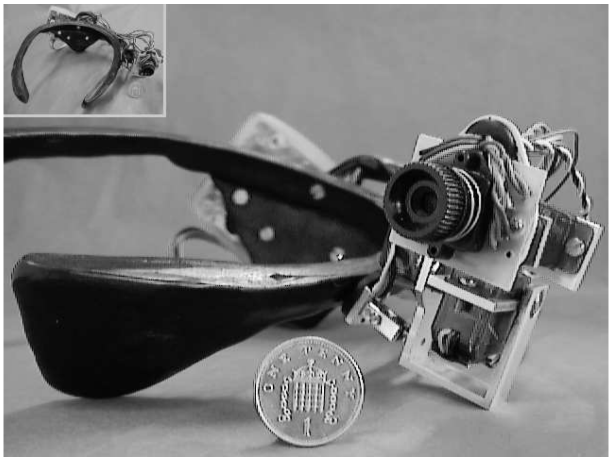


Figure 2. Collar-mounted wearable robot equipped with a miniature camera manipulated by a three-axis motorized active platform.

vide localisation information, achieving localisation using only the image data acquired from the robot's own camera in natural scenes is very appealing: the robot is making use of the same data type as its human wearer and potential remote collaborator, and is able directly to make use of measurements of the objects of interest themselves. Visual annotation for augmented reality becomes natural and accurate, and restrictions on user movement are removed.

Ego-motion estimation for an agile single camera moving through general, unknown scenes is a very challenging problem when **real-time** performance is required — most successful structure from motion work has been achieved under the off-line processing conditions. This task of estimating camera motion from measurements of a continuously expanding set of self-mapped visual features is one of a class of problems known as Simultaneous Localisation and Mapping (SLAM) in the robotics community.

Large scale real-time visual mapping of whole rooms, buildings or even outdoor scenes is currently unfeasible: as we shall see, the computational complexity of SLAM grows with map size and this means that the hard real-time constraints imposed by the requirement for continuous localisation will be violated at some map size. We are therefore currently focused on real-time small-scale **workspace localisation**. A workspace may be the volume of several cubic metres in which a wearer must carry out a task.

Localisation within a workspace entails solving a series of problems imposed by proximity. When objects are close, perspective changes and objects' mutual occlusions become pronounced and frequent even under moderately small motions of the camera. Furthermore, objects can change position and orientation or even disappear, perhaps moved by the wearer himself. Another relevant issue is that relatively small objects like hands, can inflict large amounts of occlusion because they inhabit the space near the sensor. Our method deals with these issues by mapping a widely-spaced, sparse set of features to act as localisation landmarks. While it cannot cope with full scene occlusion (an issue that would be helped greatly by the use of a wide-angle lens, at the cost of some angular resolution), partial scene occlusion and periods when only small numbers of features are visible are dealt with naturally.

2 A Wearable Robot with Active Vision

At the human scale, a moving lens-camera system is a good compromise between high angular resolution and small volume. Both criteria are of great importance for living beings and thus, not surprisingly, important to any wearable agent that browses the world using a visual sensor. Depending on its extent, robot motion can be used to compensate for user posture and motion changes or even perhaps to access places occluded in the line of sight from an otherwise fixed sensing location.

We have developed a miniature wearable active vision system (Figure 2) which in its most recent version incorporates an IEEE-1394 camera with a SONY Wfine* CCD, and field of view (FOV) of about 40° horizontal and 30° vertical. It is worn at the shoulder as a compromise between large FOV (minimising occlusion by the body) and movement independence from the wearer's viewing direction [14].

The robot has three rotational degrees of freedom (elevation, pan and cyclotorsion), in a configuration that was optimised to minimise working volume [13]. It also has a two-dimensional accelerometer but this was not used in the current work. Its controller may operate via a wireless or umbilical connection to the host computer.

We think of this platform as an interface between computer and wearer. Other authors have used wearable active vision for face tracking [10], and to sense the specific region that the wearer is attending with his eyes [16].

3 Single Camera SLAM

In this and the following sections we present our general approach to single camera localisation, valid whether the camera is worn as in the application presented in this paper, waved in the hand or even attached to a robot. More general information on this approach can be found in [4]. In our approach to visual localisation, the goal is not the processing of image sequences received from an external source, but the real-time use of a wearable camera in context. Within a room, the camera starts approximately at rest with some known object in view to act as a starting point and provide a metric scale to the proceedings (this can be as simple as a standard piece of paper). The camera then moves smoothly but rapidly, translating and rotating freely in 3D, within the room or a restricted volume within it, such that various parts of the unknown environment come into view. The aim is to estimate its 3D position continuously, promptly and repeatably during arbitrarily long periods of movement. This will involve accurately mapping (estimating the locations of) a sparse set of features in the environment.

A key aspect of our scenario is the desire for **repeatable** localisation: by this we mean requiring the ability to estimate the location of the camera with just as much accuracy after 10 minutes of motion as was possible after 10 seconds — a gradual drifting over time is not acceptable. To achieve this the features detected and mapped must function as stable, long-term **landmarks** rather than transient tracking points, and this implies both that the features must be strongly salient and identifiable, and that care must be taken when propagating the uncertainty in their locations. Early implementations of sequential structure from motion [1, 9, 2] used the standard short-lived "corner" features familiar from off-line methods and independent estimators for the location of each feature, and displayed significant motion drift over time: the inability either to re-recognise features from the past or make correct use of measurements meant that the trajectories and maps estimated displayed a gradual divergence over time from the fiducial coordinate frame.

3.1 SLAM with First-Order Uncertainty Propagation

The question of motion drift in real-time simultaneous localisation and mapping (SLAM) is now well-understood in mobile robotics research. Extended Kalman Filter (EKF)-based algorithms, propagating first-order uncertainty in the coupled estimates of robot and map feature positions, combined with various techniques for reducing computational complexity in large maps, have shown great success in enabling robots to estimate their locations accurately and robustly over large movement areas [7, 12]. In the first-order uncertainty propagation framework, the overall "state" of the system $\mathbf{x}$ is represented as a vector which can be partitioned into the state $\hat{\mathbf{x}}_v$ of the robot (or camera) and the states $\hat{\mathbf{y}}_i$ of entries in the map of its surroundings. Crucially, the state vector is accompanied by a single covariance matrix $\mathtt{P}$ which can also be partitioned as follows:

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

The role of the covariance matrix is to represent the uncertainty, to first order, in all the quantities in the state vector. Feature estimates $\hat{\mathbf{y}}_i$ can be freely added to or deleted from the map as required, $\mathbf{x}$ and $\mathtt{P}$ growing or shrinking dynamically. In normal operation, $\mathbf{x}$ and $\mathtt{P}$ change in two steps: 1. during motion, a prediction step uses a **motion model** to calculate how the robot (or camera) moves and how its position uncertainty increases; 2. when feature measurements are obtained, a **measurement model** describes how map and robot uncertainty can be reduced.

The critical importance of maintaining a full covariance matrix $\mathtt{P}$, complete with off-diagonal elements, has been irrefutably proven in SLAM research. These elements represent the correlation between estimates which is always inherent in map-building. The typical situation is that clusters of close features will have position estimates which are uncertain in the world reference frame but highly correlated with one another — their relative positions are well known. Holding correlation information means that measurements of any one of this cluster correctly affects the estimates of the others, and is the key to being able to re-visit and recognise known areas after periods of neglect.

Successful SLAM approaches have generally operated using not vision but specialised sensors such as laser range-finders, and in somewhat restricted conditions including 2D planar robot movement and/or mapping, known robot control inputs and accurately-modelled dynamics. In vision, Davison and Murray [6] made early progress in full-covariance mapping using active stereo and Davison and Kita [5], in perhaps the first work on SLAM in full 3D, used a curvature model for unknown surface shape in combination with active stereo to estimate the location of a robot moving on non-flat surfaces.

Single camera SLAM with general 3D motion is at the very difficult extreme of the genre. Among previous work, that of Chiuso *et al.*[3] has most in common with the present paper. They present a real-time, full-covariance Kalman Filter-based approach to sequential structure from motion, but aim towards model generation rather than localisation. Bottom-up 2D feature tracking means that only relatively slow camera motions are permissible, and does not allow features to be re-acquired after periods of neglect: their features typically survive for 20–40 frames then are replaced in the state vector by others. This means that motion drift would eventually enter the system.

There is much interest in real-time camera-based localisation from the wearable computing community. Foxlin [8] has demonstrated an impressive system combining accurate inertial sensing with visual measurement of automatically-mapped fiducial targets placed on a ceiling to provide real-time localisation over extended indoor areas. Kourogi *et al.* [11] also use inertial sensing in combination with visual recognition of key-framed waypoints to permit localisation-based annotation.

4 Representing 3D Position and Orientation

We define the coordinate frames W, fixed in the world, and R, fixed with respect to the camera (see Figure 3). To ease issues with linearisation and singularities, we choose a non-minimal representation of 3D orientation, and use a quaternion. The vector of 7 parameters chosen to represent position and orientation is therefore:

$$\mathbf{x}_p = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \end{pmatrix} = \begin{pmatrix} x & y & z & q_0 & q_x & q_y & q_z \end{pmatrix}^\top$$

We refer to $\mathbf{x}_p$ as the **position state** of the camera: a standard way to define 3D position and orientation which is common for any type of moving body. We differentiate between $\mathbf{x}_p$ and $\mathbf{x}_v$, the actual **state** of the body, which may include parameters additional to those representing pure position such as velocity.

5 A Motion Model for a Smoothly Moving Camera

In the case of a camera attached to a person, the motion model must take account of the unknown intentions of the person, but these can be statistically modelled. The type of model we choose initially is a “constant velocity, constant angular velocity model”. This means not that we assume that the camera moves at a constant velocity over all time, but that our statistical model of its motion in a time step is that on average we expect its velocity and angular velocity to remain the same, while undetermined accelerations occur with a Gaussian profile. The implication of this model is that we are imposing a certain smoothness on the camera motion: very large accelerations are relatively unlikely. Modelling the velocity of the camera in this way means that we must augment the position state vector $\mathbf{x}_p$ with velocity terms to form the state vector:

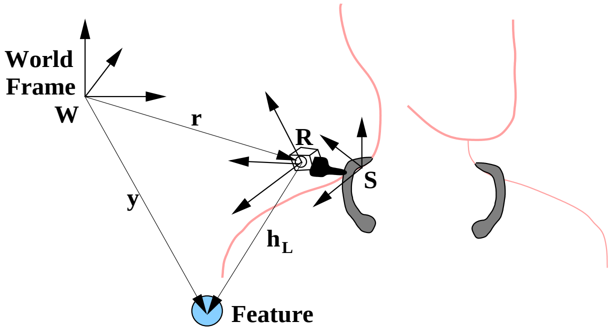


Figure 3. Frames and vectors in camera geometry. S is the shoulder frame, fixed with respect to the wearer, and R the robot frame is fixed with respect to the camera.

$$\mathbf{x}_v = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \\ \mathbf{v}^W \\ \omega^W \end{pmatrix} .$$

Here $\mathbf{v}^W$ is the linear velocity and ω^W the **angular velocity**. Angular velocity is a vector whose orientation denotes the axis of rotation and whose magnitude the rate of rotation in radians per second. The total dimension of the camera state vector is therefore 13. (Note that the redundancy in the quaternion part of the state vector means that we must perform a normalisation at each step of the EKF to ensure that each filtering step results in a true quaternion satisfying $q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1$; this normalisation is accompanied by a corresponding Jacobian calculation affecting the covariance matrix.)

We assume that in each time step, unknown acceleration $\mathbf{a}^W$ and angular acceleration α^W processes of zero mean and Gaussian distribution cause an impulse of velocity and angular velocity:

$$\mathbf{n} = \begin{pmatrix} \mathbf{V}^W \\ \mathbf{\Omega}^W \end{pmatrix} = \begin{pmatrix} \mathbf{a}^W \Delta t \\ \alpha^W \Delta t \end{pmatrix}$$

to be applied to the camera. Depending on the circumstances, $\mathbf{V}^W$ and $\mathbf{\Omega}^W$ may be coupled together (for example, by assuming that a single force impulse is applied to the rigid shape of the body carrying the camera at every time step, producing correlated changes in its linear and angular velocity). Currently, however, we assume that the covariance matrix of the noise vector $\mathbf{n}$ is diagonal, representing uncorrelated noise in all linear and rotational components. The state update produced is:

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r}^W_{new} \\ \mathbf{q}^{WR}_{new} \\ \mathbf{v}^W_{new} \\ \omega^W_{new} \end{pmatrix} = \begin{pmatrix} \mathbf{r}^W + (\mathbf{v}^W + \mathbf{V}^W)\Delta t \\ \mathbf{q}^{WR} \times \mathbf{q}((\omega^W + \mathbf{\Omega}^W)\Delta t) \\ \mathbf{v}^W + \mathbf{V}^W \\ \omega^W + \mathbf{\Omega}^W \end{pmatrix} .$$

Here the notation $\mathbf{q}((\omega^W + \mathbf{\Omega}^W)\Delta t)$ denotes the quaternion trivially defined by the angle-axis rotation vector $(\omega^W + \mathbf{\Omega}^W)\Delta t$.

In the EKF, the new state estimate $\mathbf{f}_v(\mathbf{x}_v, \mathbf{u})$ must be accompanied by the increase in state uncertainty (process noise covariance) $\mathtt{Q}_v$ for the camera after this motion. We find $\mathtt{Q}_v$ via the Jacobian calculation:

$$\mathtt{Q}_v = \frac{\partial \mathbf{f}_v}{\partial \mathbf{n}} \mathtt{P}_n \frac{\partial \mathbf{f}_v}{\partial \mathbf{n}}^\top ,$$

where $\mathtt{P}_n$ is the covariance of noise vector $\mathbf{n}$. This Jacobian calculation is complicated but tractable; we do not present the results here.

The rate of growth of uncertainty in this motion model is determined by the size of $\mathtt{P}_n$, and setting these parameters to small or large values defines the smoothness of the motion we expect. With small $\mathtt{P}_n$, we expect a very smooth motion with small accelerations, and would be well placed to track motions of this type, but would not be able to cope with sudden rapid movements. High $\mathtt{P}_n$ means that the uncertainty in the system increases significantly at each time step, and while this gives the ability to cope with rapid accelerations the very large uncertainty means that a lot of good measurements must be made at each time step to constrain estimates.

6 Visual Feature Measurements

We have followed the approach of Davison and Murray [6], who showed that relatively large (9×9 to 15×15 pixels) image patches are able to serve as long-term landmark features with a surprising degree of viewpoint-independence (see Figure 4(a)). Each interest region is detected once with the saliency operator of Shi and Tomasi [15], and matched in subsequent frames using normalised sum-of-squared difference correlation.

In this section we consider the **measurement model** of the process of measuring a feature already in the SLAM map. First, the estimates $\mathbf{x}_p$ of camera position and $\mathbf{y}_i$ (a straightforward 3D position vector) of feature position allow the value of this measurement to be **predicted**. Considering the vector sum of Figure 3, the position of a point feature relative to the camera is expected to be:

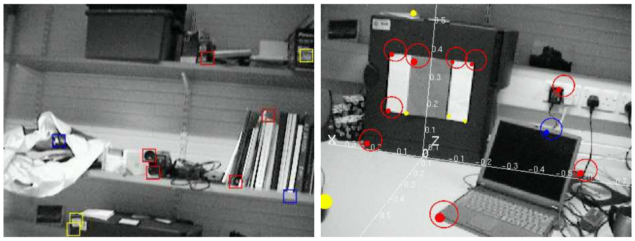

(a) (b)

Figure 4. (a) Feature image patches. Patches are detected as in [15] and generally correspond to well-localised point objects, though reflections or depth discontinuities can throw up unsuitable candidates: in SLAM, these can be rejected over time since they do not behave as stationary landmarks when observed from many viewpoints. (b) Search regions during a period of high acceleration: the positions at which features are found (small ellipses representing estimates after filtering) lie towards the boundary of the large search ellipses.

$$\mathbf{h}^R_L = \mathtt{R}^{RW}(\mathbf{y}^W_i - \mathbf{r}^W) .$$

$\mathtt{R}^{RW}$ is the rotation matrix transforming between from camera frame R and world frame W. The position (u, v) at which the feature is expected to be found in the image can be found using the standard pinohle camera model:

$$\mathbf{h}_i = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - f k_u \frac{h^R_{Lx}}{h^R_{Lz}} \\ v_0 - f k_v \frac{h^R_{Ly}}{h^R_{Lz}} \end{pmatrix} .$$

Here f is the camera focal length and k_u and k_v are CCD pixel densities in the horizontal and vertical directions respectively. Further, however, we can also calculate the uncertainty in this prediction, represented by the innovation covariance matrix $\mathtt{S}_i$. Knowledge of $\mathtt{S}_i$ is what permits an active approach to image search; $\mathtt{S}_i$ represents the shape of a 2D Gaussian pdf over image coordinates and choosing a number of standard deviations (gating, normally at 3σ) defines an elliptical search window within which the feature should lie with high probability. In our system, correlation searches always occur within gated search regions, maximising efficiency and minimising the chance of mismatches. See Figure 4(b).

$\mathtt{S}_i$ has a further role in active search: it is a measure of the information content expected of a measurement. Essentially, feature searches with high $\mathtt{S}_i$ (where the result is difficult to predict) will provide more information and produce

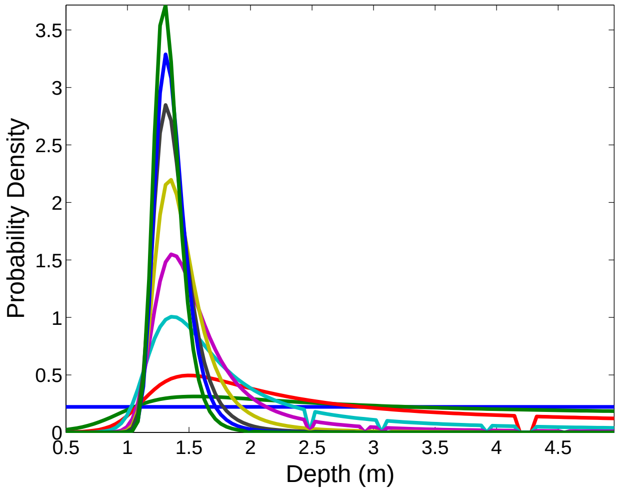


Figure 6. Frame-by-frame evolution of the probability density over feature depth represented by a particle set. 100 equally-weighted particles are initially spread evenly along the range 0.5m to 5.0m; with each subsequent image measurement the distribution becomes more peaked and more closely Gaussian.

more reduction in the uncertainty in estimates of the camera and feature positions. Specifically, since S_i depends on the uncertainty in the relative position of the camera and a feature, choosing the features with high S_i for measurement acts to reduce the uncertainty in the map consistency as a whole (always attempting to squash the multi-dimensional uncertainty in the system along the widest dimension possible). In an implementation of vision-based SLAM for a robot with steerable cameras [6] this led directly to active control of the viewing direction towards profitable measurements; here we currently do not directly control the camera movement to select valuable feature measurements but in the case that many candidate measurements are available we select those with high innovation covariance. Choosing measurements like this aims to squash the uncertainty in the system along the longest axis available at each step, and helps to ensure that no particular component of uncertainty in the estimated state gets out of hand.

7 Automatic Feature Initialisation Using Factored Sampling

The projective nature of camera measurements means that while our measurement model tells us the value of an image measurement given the position of the camera and a feature, it cannot be directly inverted to give the position of a feature given an image measurement and camera position since the feature depth is unknown. This means that initialising features in single camera SLAM will be a difficult task: initial 3D positions for features cannot be estimated from one measurement alone.

An obvious way to initialise features would be to track them in 2D in the image over a number of frames and then perform a mini-batch update when enough evidence had been gathered about their depth. However, this would violate our top-down methodolgy and waste available information: such 2D tracking is actually very difficult when the camera is potentially moving fast. Additionally, we will commonly need to initialise features very quickly because a camera with a narrow field of view may soon pass them by.

The approach we therefore take is to initialise a 3D line into the map from the single measurement, along which. the feature must lie. This is a semi-infinite line, starting at the estimated camera position and heading to infinity along the feature viewing direction, and like other map members has Gaussian uncertainty in its parameters. Its representation in the SLAM map is: $\mathbf{y}_{pi} = \begin{pmatrix} \mathbf{r}_i^W \\ \hat{\mathbf{h}}_i^W \end{pmatrix}$ where $\mathbf{r}_i$ is the position of its finite end and $\hat{\mathbf{h}}_i^W$ is a unit vector describing its direction. Along this line, a set of discrete depth hypotheses are made, analogous to a 1D particle distribution: currently, the prior probability used is uniform with 100 particles in the range 0.5m to 5.0m, reflecting indoor operation. At subsequent time steps, these hypotheses are all tested by projecting them into the image. As Figure 5 shows, each particle translates into an elliptical search region. Feature matching within each ellipse (via an efficient implementation for the case of search multiple overlapping ellipses for the same image patch) produces a likelihood for each, and their probabilities are reweighted. During the time that the particle depth distribution is being refined, the parameters of the line are not updated (except via their indirect coupling to the robot state in the Kalman Filter), and measurements of it are not used to update the camera position estimate. This is of course an approximation because in principle measurements of even a partially initialised feature do provide some information on the camera position.

Figure 6 shows the evolution of the depth distribution over time, from uniform prior to sharp peak. When the ratio of the standard deviation of depth and the depth estimate itself drops below a threshold, the distribution is safely approximated as Gaussian and the feature initialised as a point into the map — from this point onwards it behaves as a normal point feature, contributing to the update of the camera position estimate. Typically a just-initialised feature will still have a relatively large depth uncertainty (of the order of a few tens of centimetres), but this is rapidly reduced once more measurements are obtained.

The important factor of this initialisation is the shape of

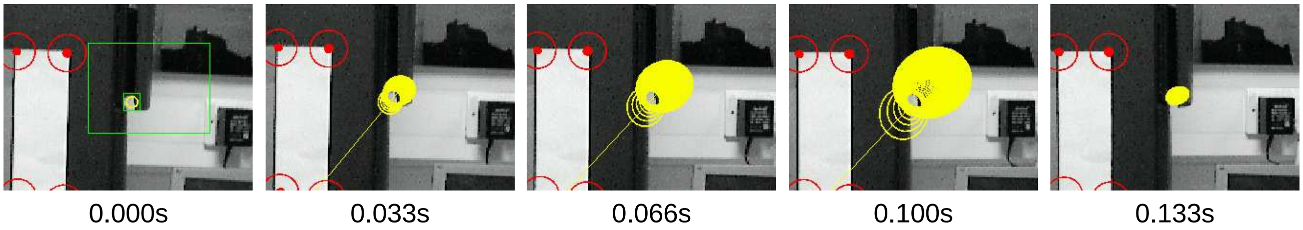


Figure 5. A close-up view of image search in successive frames during feature initialisation. In the first frame a candidate feature image patch is identified within a region searched with an interest operator. A 3D ray along which the feature must lie is added to the SLAM map, and this ray is projected into subsequent images (an epipolar line). A distribution of depth hypotheses from 0.5m to 5m translates via the uncertainty in the new camera position relative to the ray into a set of ellipses which are all searched to produce likelihoods for Bayesian re-weighting of the depth distribution. A small number of time-steps are normally sufficient to reduce depth uncertainly sufficiently to approximate as Gaussian and enable the feature to be added for use in the SLAM map.

the search regions generated by the overlapping ellipses. A simple depth prior has removed the need to search along the entire epipolar line, and improved the robustness and speed of initialisation. In real-time implementation, the speed of collapse of the particle distribution is aided (and correlation search work saved) by deterministic pruning of the weakest particles at each step.

7.1 Map Management

With the ability to add features to the map comes the need for criteria to decide when this should be necessary, and potentially when some features should be deleted. Our map-maintenance criterion aims to keep the number of reliable features visible from any camera location close to a pre-determined value determined by the specifics of the measurement process, the required localisation accuracy and the computing power available: currently, numbers in the region 6–10 are used in this work. Feature "visibility" (more accurately predicted measurability) is calculated based on the relative position of the camera and feature, and the saved position of the camera from which the feature was initialised: the feature must be predicted to lie within the image (and not too close to the edge), but further the camera must not have moved or rotated too far from its initialisation viewpoint of the feature or we would expect correlation to fail. Features are added to the map if the number visible in the area the camera is passing through is less than this threshold. 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. 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%) are failures. This criterion prunes "bad" features which are not true 3D points or are often occluded.

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

8 Active Camera Control: Saccade and Pursuit

The capability for active robotic control of the orientation of the wearable camera really comes into its own when combined with the detailed real-time localisation information available from visual SLAM. In previous work [14], a roll/pitch accelerometer sensor mounted on the wearable robot permitted self-levelling orientation control of the camera with respect to gravity. Full 3D localisation however also allows control of the camera based on position information — for instance it enables:

- Extended fixation on a 3D object during wearer motion.

- Controlled saccades between known objects in arbitrary positions, even when the target is out of the current field of view.

These are capabilities which were certainly impossible using only orientation sensing. Once a map of various features has been built up, the robot can be directed at will to any mapped feature and commanded to control its orientation to maintain fixation during an extended period of user motion. Continuous tracking of a single feature is something which could be achieved in simpler ways, using a visual servoing approach. In our scheme, however, since global localisation is continuously recovered, a further command can then send the robot back to fixate any **other** known feature: the camera position estimate, feature position estimate and knowledge of the current robot angles permit immediate calculation of the control demand necessary for fixation.

8.1 Control Scheme

A single simple control rule is used to calculate robot orientation demands during feature fixation tracking and inter-feature saccades. First, as when making feature measurements, the vector from the camera centre to the feature on which fixation is desired is calculated and transformed into the camera frame of reference R.

$$\mathbf{h}_L^R = \mathtt{R}^{RW}(\mathbf{y}_i^W - \mathbf{r}^W) \ .$$

Knowledge via odometry of the current wearable robot motor angles permits calculation of the rotation matrix $\mathtt{R}^{SR}$ transforming between the camera frame R and the shoulder frame S; using this (and assuming that the offsets in the robot geometry are small such that the camera rotates approximately about its optic centre), we calculate the camera-feature vector in the shoulder frame:

$$\mathbf{h}_L^S = \mathtt{R}^{SR}\mathbf{h}_L^R \ .$$

This vector can then be decomposed to determine the ideal robot angles (elevation and pan) for direct fixation on the feature. These are not demanded straight away however — this would lead to very fast camera motions during which tracking would likely be lost. Rather, a maximum angular velocity limit is fixed for each axis (currently at $30°s^{-1}$) and this defines the maximum demand increment issued at any time. Simultaneously the cyclotorsion degree of freedom is controlled such that the camera remains maximally horizontal (with respect to the world coordinate frame defined by the initially known features in the map). This means that the camera will stay fairly horizontal during the wearer's twists and turns, aiding feature matching and tracking.

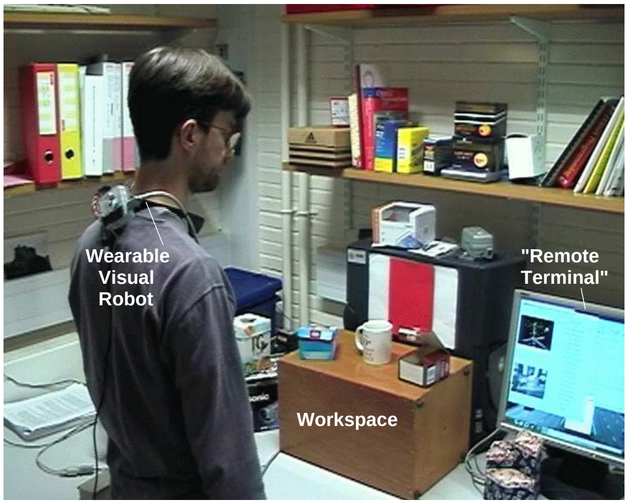


Figure 7. Experimental environment.

9 Results

The visual SLAM algorithm was implemented at full 30Hz operation. In the current implementation all vision and map processing was achieved on a lap-top with a 1.6GHz Centrino processor. In fact, processing at a rate of less than 30Hz would no doubt be difficult since the inter-frame motions induced may be very great and the uncertainty required of the motion model such that all search regions would be very large — there is indeed a powerful argument for proceeding to processing rates higher than 30Hz, where the active approach used means that less processing would be required on each images since search regions would be smaller reflecting reduced motion uncertainty.

The experimental environment is depicted in Figure 7. The wearer browses a desk-top scene, manipulating objects and moving smoothly but freely. Meanwhile a remote operator (in this case sitting nearby) manipulates the PC where output from the robot is displayed interactively.

The system is able routinely to keep track of localisation during long periods of several minutes of wearer motion, including often tracking through times when only very few features are visible thanks to the stability provided by the motion model — there is no minimum to the number of features that must be successfully measured at each frame although of course more is always preferable. After an extended period of movement, a typical map generated may have around 50 or 60 features and span a working volume of several cubic metres. Larger maps than this are unfeasible with the current implementation due to real-time processing constraints.

The positions of six features corresponding to corners of a paper target were given to the system as prior knowledge (image patches were selected and saved by hand, and their 3D positions relative to a defined coordinate frame accurately measured — these features are inserted into the SLAM map with zero uncertainty, and therefore all rows and columns of the covariance matrix relating to them will always have zero values). The initial position $\mathbf{x}_v$ of the camera within this coordinate frame was also measured, though this estimate is inserted into the state vector accompanied by a covariance $\mathtt{P}_{xx}$ which corresponding to an uncertainty of a few centimetres and this enables tracking to start as long as the initial camera position is reasonably close to that defined. It would of course be desirable to be able to start tracking without the need for a known target, but currently this seems unfeasible: the target defines a known length scale for the system without which progress would be difficult since our scheme makes use of metric priors in the motion model and initialisation process.

Linear acceleration noise components in $\mathtt{P}_n$ were set to a standard deviation of 1ms^{-2}, and angular components with a standard deviation of 6rads^{-2}. The relatively large angular term was necessary to cope with the rapid changes in orientation of a worn camera.

9.1 Interface for Collaboration with a Remote Expert

Two real-time graphical displays are presented to the remote expert: a three-dimensional reconstruction of the estimated locations of the camera and features, and the image view from the camera augmented with feature estimates, a world coordinate frame and other graphics. The remote expert's role in collaboration is to click on the displays to highlight objects for the wearer's attention. It is actually very difficult for the remote collaborator to click reliably on objects in the rapidly-moving camera view — the stable 3D view in the world coordinate frame proves valuable for this, though more detailed graphics than the set of points currently drawn for rendering speed would be desirable.

9.2 Video

A video accompanying this paper is available from our website (see figure 8). This presents the results achieved in the best manner and shows different views of real-time SLAM, wearable robot control and annotation. In graphical views, the feature colours used indicate their status: red features have been measured at the current time-step, blue indicates that an attempted measurement has failed, and yellow that a measurement was not attempted (the feature having been evaluated as not measurable from this position or because enough better features have already been selected). A feature currently being used as the target for fixation is drawn in green, and should consistently appear close to the image centre. Simple view augmentation with wire-frame cubes centred at the location of a selected feature highlights certain positions. The cubes' orientations are fixed relative to the world coordinate frame and they have constant 3D size.

10 Conclusions

We have presented a fully-automatic real-time visual localisation and mapping system which is a very important step towards useful wearable vision and relevant to all types of camera motion estimation. Current limitations of the algorithm include that it can only handle a certain number of features (perhaps 60 with full graphical output, or 100 without) within real-time processing constraints. This is enough to span a workspace, but not a whole room. Efficient SLAM implementations and improved feature matching are required to improve this.

The feature matching would benefit from more viewpoint-independent characteristics; in particular it cannot cope with too much camera rotation about the optic axis although with the wearable robot cyclotorsion control mitigates this. We are currently investigating 3D surface patches as a replacement for the current 2D features.

The narrow field of view of the wearable is a problem for the SLAM algorithm, in which it is desirable to see a wide spread of features at all times. A more wide-angle lens should be beneficial since fewer features will need to be added to the map.

We do not currently make use of any inertial sensing, although incorporating this would be a natural step. The inevitable increase in performance would need to be weighed against the small increase in system complexity involved, as well as the loss of algorithmic generality: a purely vision-based method is more readily applied to different hardware platforms.

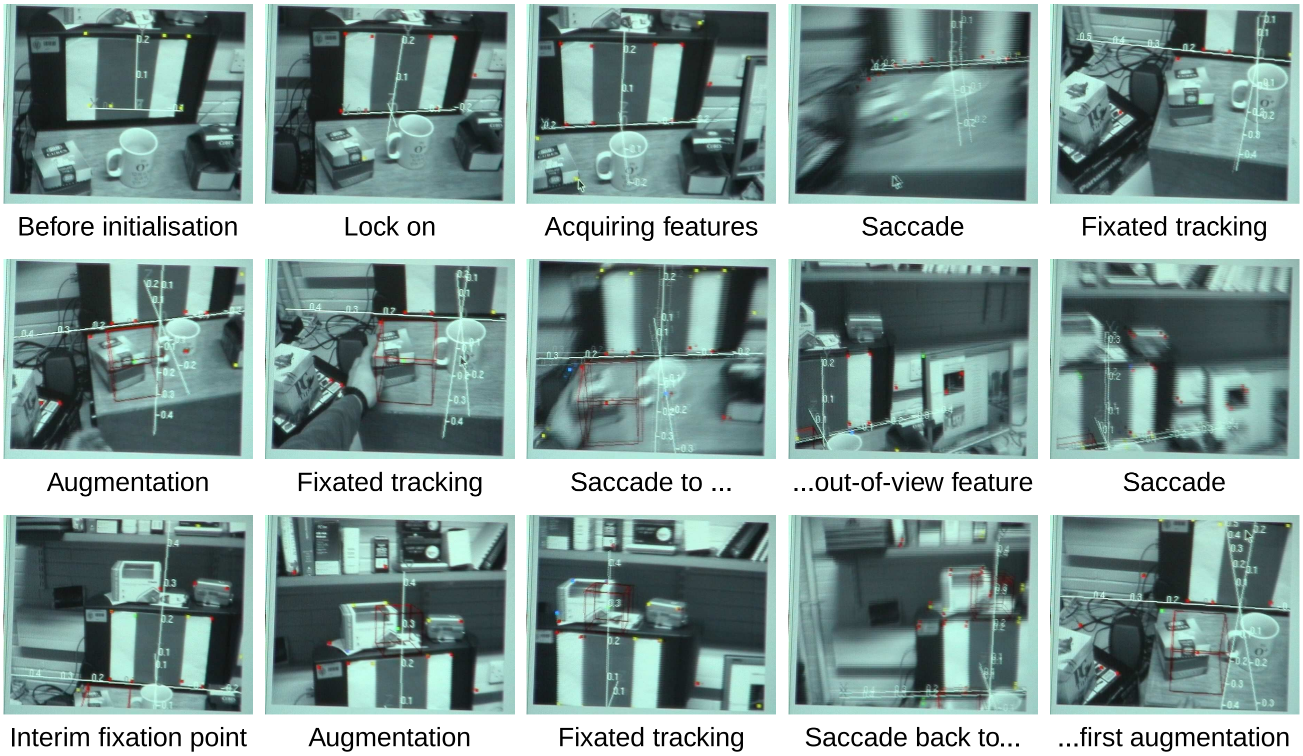
Current plans are to move the system away from the desktop to more general large-scale scenes.

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 *Proceedings of the 5th International Conference on Computer Vision, Boston*, pages 58–65. IEEE Computer Society Press, 1995.

[3] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. "MFm": 3-D motion from 2-D motion causally integrated over time. In *Proceedings of the 6th European Conference on Computer Vision, Dublin*, 2000.



Before initialisation | Lock on | Acquiring features | Saccade | Fixated tracking

Augmentation | Fixated tracking | Saccade to ... | ...out-of-view feature | Saccade

Interim fixation point | Augmentation | Fixated tracking | Saccade back to... | ...first augmentation

Figure 8. A story-board selection of frames from the video accompanying this paper showing tracking of features and saccades between them, all during continuous wearer motion. Note that the blurring in the saccade frames is an artifact of the video making process. Video available at `http://www.robots.ox.ac.uk/ActiveVision/Movies/wearableslam.mpg`

[4] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the 9th International Conference on Computer Vision, Nice*, 2003.

[5] A. J. Davison and N. Kita. 3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. 2001. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[6] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002.

[7] H. F. Durrant-Whyte, M. W. M. G. Dissanayake, and P. W. Gibbens. Toward deployments of large scale simultaneous localisation and map building (SLAM) systems. In *Proceedings of the 9th International Symposium of Robotics Research, Snowbird, Utah*, pages 121–127, 1999.

[8] E. Foxlin. Generalized architecture for simultaneous localization, auto-calibration and map-building. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 2002.

[9] C. G. Harris. Geometry from visual motion. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, Cambridge, MA, 1992.

[10] T. Kato, T. Kurata, and K. Sakaue. Face registration using wearable active vision systems for augmented memory. In *Proc. Digital Image Computing: Techniques and Applications (DICTA2002)*, Melbourne, Australia, 2002.

[11] M. Kourogi, T. Kurata, and K. Sakaue. A panorama-based method of personal positioning and orientation and its real-time applications for wearable computers. In *Proc. ISWC*, pages 107–114, 2001.

[12] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Robotics Research*. Springer Verlag, 2000.

[13] W. Mayol, B. Tordoff, and D. Murray. Designing a miniature wearable visual robot. In *IEEE Int. Conf. on Robotics and Automation*, Washington DC, USA, 2002.

[14] W. W. Mayol, B. Tordoff, and D. W. Murray. Wearable visual robots. In *IEEE International Symposium on Wearable Computers*, Atlanta GA, USA, 2000.

[15] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[16] A. Sugimoto and T. Matsuyama. Detecting person's blink points and estimating human motion trajectory. In *The First Int. Workshop on Man-Machine Symbiotic Systems*, Kyoto, Japan, 2002.