Real-Time Simultaneous Localisation and Mapping with a Single Camera

Andrew J. Davison
Robotics Research Group, Dept. of Engineering Science, University of Oxford, OX1 3PJ, UK
ajd@robots.ox.ac.uk
http://www.robots.ox.ac.uk/~ajd/

Abstract

Ego-motion estimation for an agile single camera moving through general, unknown scenes becomes a much more challenging problem when **real-time** *performance is required rather than under the off-line processing conditions under which most successful structure from motion work has been achieved. 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, and we argue that such real-time mapping research, despite rarely being camera-based, is more relevant here than off-line structure from motion methods due to the more fundamental emphasis placed on propagation of uncertainty.*

We present a top-down Bayesian framework for single-camera localisation via mapping of a sparse set of natural features using motion modelling and an information-guided active measurement strategy, in particular addressing the difficult issue of real-time feature initialisation via a factored sampling approach. Real-time handling of uncertainty permits robust localisation via the creating and active measurement of a sparse map of landmarks such that regions can be re-visited after periods of neglect and localisation can continue through periods when few features are visible. Results are presented of real-time localisation for a hand-waved camera with very sparse prior scene knowledge and all processing carried out on a desktop PC.

1 Introduction

A new range of applications for camera-based localisation and mapping opens up if the border to real-time processing can be crossed: with real-time processing, motion estimates for a moving camera are available for immediate use in interactive scenarios. A live camera connected to a computer becomes a real-time position sensor which could be applied with a minimum of domain knowledge to areas in robotics (motion estimation for generally moving robots such as humanoids), wearable robotics (motion estimation for camera-equipped devices worn by humans), telepresence (head motion estimation using an outward-looking camera), or television (camera motion estimation for live augmented reality).

Structure from motion research in computer vision has reached the point where fully automated reconstruction of the trajectory of a video camera moving through a previously unknown scene is becoming routine [8], but these and other successful approaches seen to date have been formulated as **off-line** algorithms and required **batch**, simultaneous processing of all the images acquired in the sequence. In these methods, large numbers of well-localised features of high image interest (usually "corner" points or lines) are detected in each image of a video sequence and, postulating that each is associated with a repeatably identifiable 3D entity in the environment, matched between consecutive (or close) video frames. The assumption of rigidity in the scene is then used to assert that the feature image motion observed is due purely to the movement of the camera relative to the unknown but static 3D geometry or "structure" of the features, and this permits solutions for both the motion of the camera between the matched positions and the locations of the 3D features to be obtained. Long chains of these frame-to-frame motion solutions can be stitched together to produce an estimate of a complete camera trajectory and full 3D map of all the features observed, and the quality of the overall solution can be refined by further constrained alteration ("bundle adjustment").

Of course batch processing provides the most accurate and robust solution to any estimation problem in applications where off-line operation is satisfactory. Real-time operation, however, enforces hard constraints on the processing permissible: specifically, in the common case in which data arrives at a constant rate (e.g. from a camera at 30Hz) the estimation must operate in **constant time**, requiring an amount of processing bounded by a constant to take account of the pertinent information available from each image. The value of this constant will depend on the details of the processor available but significantly the processing time **per image** cannot increase indefinitely with time otherwise a point will always be reached at which the real-time constraint is breached. Rather than storing a history of measurements we are led towards a time-independent **state-**

based representation, in which everything of interest about the system in question is captured in a snapshot of the current instant. Sequential estimation then proceeds at each time step by **updating** the state estimate due to the effect of the passage of time and any new data obtained.

2 Repeatable Localisation

Our goal is not the processing of image sequences received from an external source, but the real-time use of a camera in context. The scenario under current consideration involves a live camera module connected to a PC (in this case via the IEEE1394 "Firewire" bus). Within a room, the camera starts 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 an A4 piece of paper). It is picked up in the hand and moved 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 arbitrary long periods of movement. This will certainly involve accurately estimating the locations of features in the environment, but in this case this mapping is considered a means to obtaining camera localisation rather than an output in itself: the goal is only to map such features as are sufficient to obtain camera localisation.

A key aspect of this 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 put into propagating the uncertainty in their locations. Early implementations of sequential structure from motion [1, 10, 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 of them meant that the trajectories estimated displayed a gradual divergence over time from the fiducial coordinate frame.

2.1 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, 13]. 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 freely be 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** described how map and robot uncertainty can be reduced.

The critical role of maintaining a full covariance matrix $\mathtt{P}$, complete with off-diagonal elements, has been irrefutably proven in SLAM research because these elements represent the correlation between estimates which is always inherent in map-building: typically clusters of close features will have position estimates which are uncertain the world reference frame but highly correlated — their relative positions are well known. Holding correlation information means that measurements of any one of this cluster correctly affects estimate 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 not using 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 [4], 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. However in both cases the algorithms were restricted to the case of smoothly moving robots with known control parameters and stereo vision.

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 as a localisation method motion drift would eventualy enter the system.

There is much interest in camera-based localisation from the wearable computing community. Foxlin [9] demonstrated an impressive system combining accurate inertial sensing with visual measurement of automatically-mapped fiducial targets on the ceiling to provide real-time localisation in extended indoor areas. Kourogi *et al.* [12] also used inertial sensing combined with visual recognition of key-framed waypoints to permit localisation-based annotation.

3 A Motion Model for a Smoothly Moving Camera

In Bayesian real-time localisation, performing estimation in an uncalibrated frame of reference would really be to lose the chance to make use of the very useful information available from a motion model and other priors. Except in some very restricted cases, motion models based in the real physics of acceleration and angular acceleration simply do not make sense in the non-metric, non-Euclidean coordinate frames often used in batch structure from motion. We therefore assume camera calibration is available and place all estimates in a world of right angles and SI units. If self-calibration was desired, given a reasonable prior guess this could be refined explicitly within the probabilistic filter (although we do not consider this issue here).

We define the coordinate frames W, fixed in the world, and R, fixed with respect to the camera (see Figure 1). 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 the **position state** $\mathbf{x}_p$.

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

Constructing a motion model for an agile camera which may for instance be carried by a person at first glance might seem to be fundamentally different to modelling the motion of a wheeled robot moving on a plane: the key difference is that in the robot case one is in possession of the **control inputs** driving the motion, such as "move forward 1m with steering angle 5°", wheras we do not have such prior information about a person's movements. However, it is important to remember that both cases are just points on the continuum of types of model for representing physical systems. Since (classical) physics is deterministic, in theory an enclosed system could be modelled down to infinitessimal precision by a collection of parameters and equations and then its future behaviour predicted for all time. In reality, however, the precision of a model always stops at some level of detail and a probabilistic assumption is made about the discrepancy between this model and reality: this is what is referred to as process noise. In the case of a wheeled robot, this noise term takes account of factors such as potential wheel slippage, surface irregularities and other predominantly unsystematic effects which have not been explicitly modelled. In the case of a camera attached to a person, it takes account of the unknown intentions of the person, but these too can be statistically modelled.

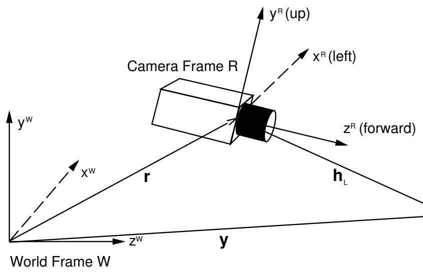

Figure 1: Frames and vectors in camera geometry.

We choose initially 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 undetermined **accelerations** occur with a Gaussian profile. The implication of this model is that we are imposing a certain smoothness on the camera motion expected: very large accelerations are relatively unlikely. This model is subtley effective and gives the whole system important robustness even when visual measurements are sparse. Let us remember that the vast majority of structure from motion methods in computer vision use no motion model at all — their approach is to throw away all information about where the camera was in previous frames and start again from scratch with each new image.

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:

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

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:

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

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 $\mathbf{n}$. EKF implementation also requires calculation of the Jacobian $\frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}$. These Jacobian calculations are complicated but a tractable matter of differentiation; 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 unable 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.

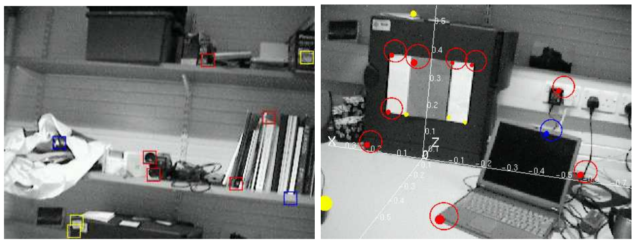
(a) (b)

Figure 2: (a) Feature image patches. Patches are detected as in [14] 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 high acceleration: the positions at which features are found (small ellipses representing estimates after filtering) lie towards the boundary of the large search ellipses.

4 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 2(a)). Each interest region is detected once with the saliency operator of Shi and Tomasi [14], 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 (we will discuss initialisation later). First, the estimates we have $\mathbf{x}_v$ 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 1, the position of a point feature relative to the camera is expected to be:

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

The position (u, v) at which the feature is expected to be found in the image isfound using the pinhole camera model:

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

Further, we can also calculate the uncertainty in this prediction, represented by the innovation covariance matrix $\mathtt{S}_i$:

$$\begin{aligned} \mathtt{S}_i \; = \; & \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v} \mathtt{P}_{xx} \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v}^\top + \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v} \mathtt{P}_{xy_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i}^\top + \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i} \mathtt{P}_{y_i x} \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v}^\top \\ & + \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i} \mathtt{P}_{y_i y_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_i}^\top + \mathtt{R} . \end{aligned}$$

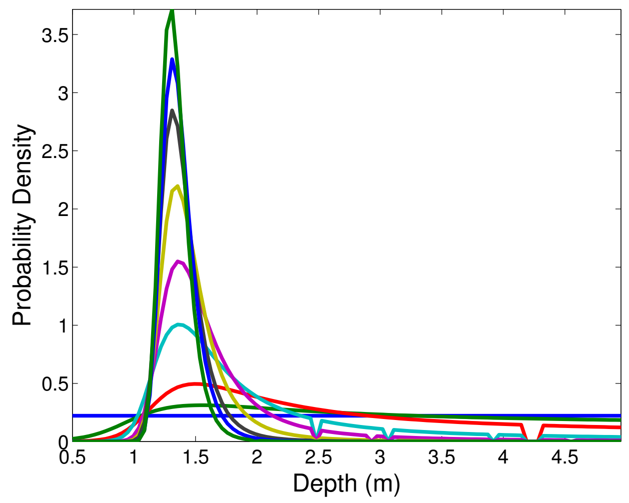

Figure 4: Frame-by-frame evolution of the probability density over feature depth represented by a particle set. 100 equally-weighted particles are initially spread evenly along the range 0.5m to 5.0m; with each subsequent image measurement the distribution becomes more closely Gaussian.

The noise covariance $\mathtt{R}$ of measurements is taken to be diagonal with magnitude determined by image resolution.

Knowledge of $\mathtt{S}_i$ is what permits a fully 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 2(b).

$\mathtt{S}_i$ has a further role in active search; it is a measure of the information content expected of a measurement. Feature searches with high $\mathtt{S}_i$ (where the result is difficult to predict) will provide more information about estimates of camera and feature positions. 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 cannot control the camera movement, 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, and helps ensures that no particular component of uncertainty in the estimated state gets out of hand.

5 Automatic Feature Initialisation

The projective nature of camera measurements means that while a measurement 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 image measurement and camera position since the feature depth is unknown. Initialising features in single camera SLAM will therefore be a difficult task: 3D depths for features cannot be estimated from one measurement.

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.

It is important to realise that a statement like "not invertible" does not have real meaning in a Bayesian framework, in which everything is uncertain and we must talk about probability distributions rather than in binary statements. Even after seeing a feature only once, we can talk about a PDF for its 3D position assuming that we had some prior belief about its depth. However, to use the feature in our SLAM map we require that its 3D position PDF can reasonably be modelled as a multi-variate Gaussian and this is why we cannot initialise it fully after just one measurement. The approach we take therefore after one measurement is to initialise a 3D **line** into the map 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 paremeters. 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 one 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 (quite a different type of prior may be required in larger environments where features may be very distant or even effectively at infinity). At subsequent time steps, these hypotheses are all tested by projecting them into the image. As Figure 3 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. Figure 4 shows the evolution of the distribution over time, from uniform prior to sharp peak. When the ratio of the standard deviation of depth to depth estimate drops below a threshold, the distribution is safely approximated as Gaussian and the feature initialised as a point into the map. The important factor of this initialisation is the shape of the search regions

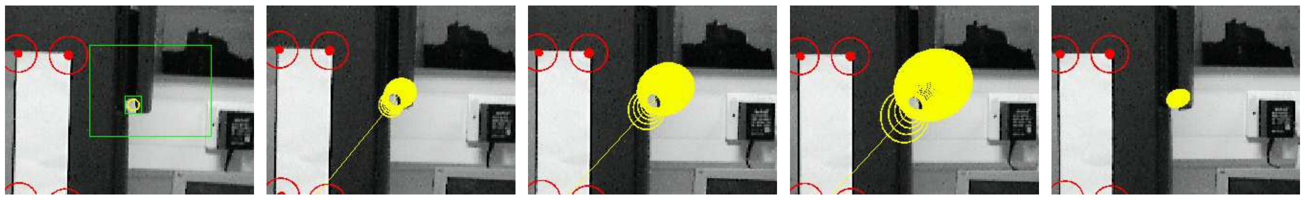

Figure 3: A close-up view of image search in successive frames during feature initialisation. In the first frame a candidate feature image patch is identified within a search region. A 3D ray along which the feature must lie is added to the SLAM map, and this ray is projected into subsequent images. 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.

generated by the overlapping ellipses. A 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.

5.1 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 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, 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 undesirable to increase the number of features and add to the computational complexity of filtering without good reason. Features are detected by running the image interest operator of Shi and Tomasi to locate the best candidate within a box of limited size (around 100×50 pixels) placed within the image — this is for reasons of efficiency in a real-time implementation. The position of the search box is currently chosen randomly, with the constraints only that it should not overlap with any existing features and that based on the current estimates of camera velocity and angular velocity any detected features are not expected to disappear from the field of view immediately. No effort is currently made to detect features in "useful" positions in terms of improving localisation information although this would be an interesting avenue for research — more important is to find the features of strong image salience which exist in the image and to have them widely distributed across the image.

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. Correct operation of the system relies on the fact that in most scenes very similar objects do not commonly appear close enough 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 repeated texture).

6 Results

We present results corresponding a video of a 20 second run of real-time SLAM (see Figure 5). 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 fea-

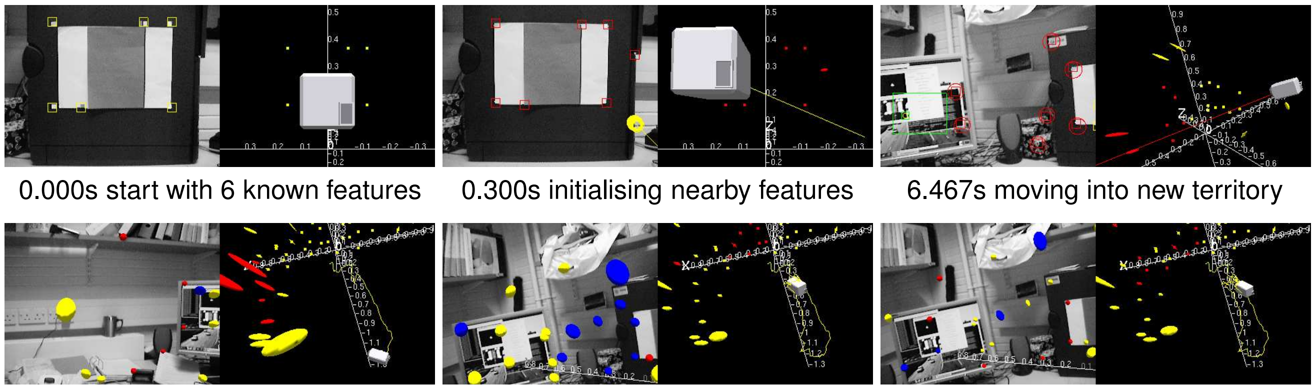
0.000s start with 6 known features
0.300s initialising nearby features
6.467s moving into new territory
18.833s sparser feature pickings
20.167s a dangerous moment
20.533s old features are re-captured

Figure 5: Frames from video illustrating a 20 second period of real-time tracking during which a map of around 60 features was built. For each time-step an image view (in which all feature estimates are projected into the estimated camera frame) is shown next to an external 3D view in which the camera can also be seen. Features are colour-coded as follows: red = successfully measured at this frame; blue = failed measurement; yellow = not selected for measurement. At different time-steps, feature location uncertainties, search regions or image patches are displayed. Video available from `http://www.robots.ox.ac.uk/~ajd/Movies/realtime_30fps_slam.mpg`.

tures 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 corresponds 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. Linear acceleration noise components in $\mathtt{P}_n$ were set to a standard deviation of 4ms^{-1}, and angular components with a standard deviation of 6rads^{-1}. These figures are sufficient to track the motion of a camera waved freely but smoothly in the hand.

Figure 5 shows snapshots from the video showing both internal and external camera views in 3D. A degree of "jitter" in the camera motion is observed, reflecting the fact that real-time operation forces the number of features use to be small, but tracking is very robust and the rare feature mismatches do not cause disaster despite the current lack of an explicit "robust statistics" approach. Features are routinely recaptured after long periods of neglect as the camera moves between viewing zones with no overlap. The characteristic drift with distance from fiducial coordinate frame of SLAM is observed, but also the correction of this drift when old features are re-acquired (a group of features is seen to move en masse after this implicit re-registration). The algorithm also tracks successfully through several periods when very few features are visible; something which would be impossible without a motion model Tracking of this kind has been observed to be very repeatable and adaptable within this kind of desk-top scenario: long periods of tracking of several minutes present no problem. Significant problems with non-linearity and the Gaussian assumptions of the EKF have not been encountered but may because significant during camera trajectories which move farther from the original coordinate frame. The method has recently been applied to the localisation of a wearable visual robot [5].

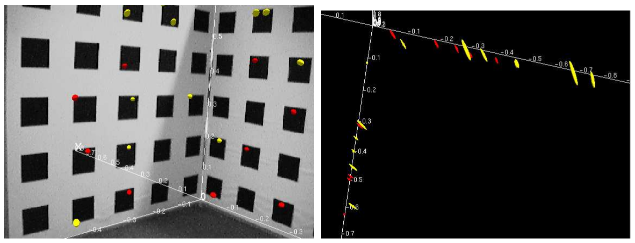

Figure 6: An experiment using an artificial scene to investigate structure recovery: real-time SLAM was carried out as the camera was waved back and forth in front of a calibration grid consisting of two highly-textured planes. Initial knowledge was of the locations of only four feature locations, lying in the left-hand plane. The images show a snapshot of the map (image and top-down external views) after 15 seconds, when much repeated measurement had reduced the feature location uncertainties to small values, and indicate that the planar surfaces have been accurately recovered.

A brief experiment depicted and described in Figure 6 investigated the absolute feature reconstruction accuracy.

6.1 Processing Time

On a 2.2GHz Pentium processor, a typical breakdown of processing time required at each frame at 30Hz (such that 33ms is available for processing each image) is as follows:

10ms	Correlation searches
5ms	Kalman Filter update
10ms	Feature initialisation search

A fundamental characteristic of full-covariance EKF SLAM is that the computational complexity of the filter update is of order N^2, where N is the number of features in the map. The Kalman Filter update time begins to grow rapidly when the number of features approaches 100, and going past this would require implementation of one of the many published SLAM shortcut methods (e.g. [11]).

7 Conclusions

We have described a principled, Bayesian, top-down approach to sequential Simultaneous Localisation and Mapping or Structure from Motion which takes account of the extra sources of information often neglected in batch methods to push performance past the real-time barrier, and demonstrated robust performance in an indoor scene.

There are a number of research issues which must be tackled before such a system could leave a desk-like environment and map a whole building or town in real-time, however. The image features used, being 2D patches, are limited in viewpoint-variance, and the algorithm would benefit from the use of features such as planar 3D patches whose change in image appearance could be better predicted from different viewpoints.

Lastly we are convinced of the benefits of active search based on information content, but there is much to be done to apply information theory rigorously in this domain. For instance, when measurements of several features are being made in each frame, what does a successful measurement of one tell us about where to look for the others? And what if there is a chance that that measurement was the result of incorrect data association?

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.

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

[5] A. J. Davison, W. W. Mayol, and D. W. Murray. Real-time localisation and mapping with wearable active vision. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, Tokyo*, 2003.

[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] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. European Conference on Computer Vision*, pages 311–326. Springer-Verlag, June 1998.

[9] 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.

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

[11] J. G. H. Knight, A. J. Davison, and I. D. Reid. Constant time SLAM using postponement. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 2001.

[12] 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.

[13] 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.

[14] 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.