SLAM with a Single Camera

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

May 9, 2002

Abstract

Real-time motion estimation for a generally moving, agile single camera is a particularly challenging SLAM problem, but one whose solution will lead to a host of lucrative and interesting applications in robotics, multimedia and television. We argue that mapping research in mobile robotics, despite rarely being camera-based, is more relevant when tackling this problem than recent structure from motion work in computer vision which has focused on off-line reconstruction of camera trajectories. We present a framework for EKF-based single-camera localisation and initial experimental results, and discuss current and future research issues.

1 Introduction

Real-time Simultaneous Localisation and Map-Building ("SLAM") in mobile robotics has seen great progress in recent years — to the point that some researchers are now claiming it to be a largely solved problem. 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 [1, 6, 9, 12, 5]. However, it is important to remember the somewhat restricted conditions under which these successful demonstrations have generally been achieved:

- 2D planar robot movement and/or mapping
- Known robot control inputs and accurately-modelled dynamics
- Slow or smooth robot motion
- Specialised, accurate, well-calibrated sensors
- Multiple sensors of the same or different types
- Simple, easy to map environments with unambiguous landmarks
- Large computational resources available

We argue that the gradual loosening of these restrictions is able to add almost limitless extra "difficulty" to the SLAM problem, and that new research issues will continue to arise. The various demonstrations produced so far remain valid, because a great number of useful robots are able to operate in restricted circumstances: for instance, large, expensive robots in industrial scenarios are often able to function in simple 2D environments, move slowly, and carry various high-performance sensors and powerful processors. Nevertheless, a gradual lifting of restrictions opens up a whole range of new applications for SLAM algorithms.

1.1 Cameras and SLAM

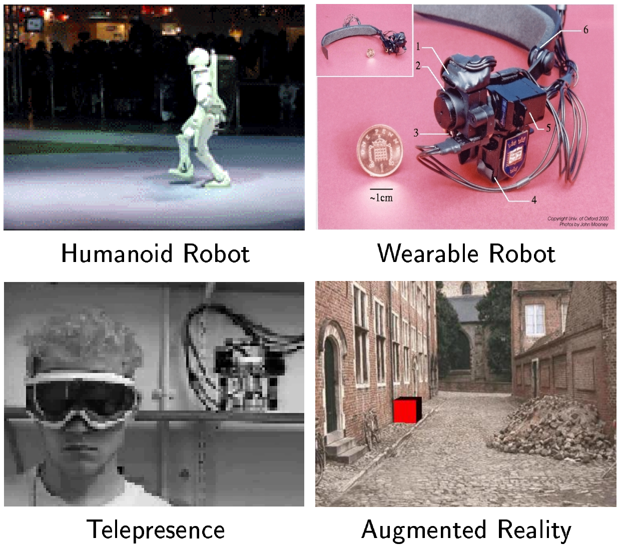
In this paper we look specifically at a SLAM problem which presents a particularly testing set of circumstances: motion estimation for a single camera, moving rapidly in 3D in normal human environments, based on mapping of visual features, potentially with minimal prior information about motion dynamics. The value in working on this problem is in the flexibility, ubiquity, compactness and power of optical cameras compared with other more esoteric sensors — reflecting the fact

Figure 1: The goal: 3D motion estimation for a generally-moving single camera.

that humans are predominantly visual animals, cameras exist in many domains.

We do not propose here that vision is necessarily the “best” sensor for SLAM: in an expensive robotic application one would likely choose a sensor like a laser range-finder first to bear the grunt of SLAM processing, and then add cameras if particular benefits could be achieved. However, in various potential applications cameras are either already present in the scenario, or could be installed very conveniently. Examples are shown in Figure 2: in the short term, camera-based SLAM will be most useful in domains where the goal is to recover camera **position** in real-time, via sparse feature maps, rather then aiming to build dense visual maps as outputs themselves: the computation involoved in building dense maps is simply too great. The short term goal should be a rather general-purpose real-time position sensor, which could be rapidly and flexibly implemented with a minimun of domain knowledge in robotics (motion estimation for generally moving robots such as humanoids), wearable robotics (motion estimation for devices worn by humans to assist in tasks such as search and rescue or industrial inspection), telepresence (human head motion estimation by means of an outward-looking head-mounted camera attached to a head-mounted display), or television (used to provide camera motion estimation for on-line augmented reality).

High performance, fully digital cameras able to acquire images at 640 × 480 pixels resolution and transfer them to any PC or laptop at 30Hz (in this case over the IEEE1394 “firewire” bus) are now available for just over US$100. Algorithms which work with cameras like these could really bring SLAM to the desktop and mean that applications could reach millions of users rather than hundreds.



Humanoid Robot

Wearable Robot

Telepresence

Augmented Reality

Figure 2: Potential applications: motion estimation for a humanoid robot (the Honda P3) or wearable robot (developed by Walterio Mayol and David Murray at the University of Oxford); head motion estimation for telepresence (picture shows robotic head slaving developed by Jason Heuring and David Murray at the University of Oxford); camera motion estimation for real-time augmented reality (picture shows off-line implementation by Kurt Cornelis and Marc Pollefeys at K.U. Leuven).

1.2 Structure from Motion

Attempting SLAM using vision brings into sharp focus the similarities between work on map-building in mobile robotics and “structure from motion” research in computer vision, where 3D models and camera trajectories are recovered from image sequences. The key goals are clearly the same: simultaeous reconstruction from sensor measurements both of the motion of the sensor body and its movement. Nevertheless, structure from motion research has taken a very different route from the methods commonly used in SLAM for a single key reason: the lack of hard real-time constrants in many useful applications for vision technology.

Structure from motion research in computer vision has reached the point where fully automated reconstruction of the trajectory of a camera and the locations of the arbitrary features it observes is becoming routine [7, 10]; however the successful approaches seen to date have almost exclusively required off-line, **batch** processing of the images acquired, via computationally costly simultaneous analysis of all the images obtained in a sequence using non-linear minimisation techniques. These off-line methods are readily applied to building 3D models from video sequences for use in video games, or for recovering camera trajectories for augmented reality effects in cinematic post-processing, and commercial products have been recently been released in these areas.

On the contrary, robotic applications have always required real-time performance, and therefore a **sequential** approach, where map-building and localisation proceed in a step-by-step fashion as movement occurs. Real-time applications require that with each new piece of data, its effect on estimates can be incorporated within the constant time-step available until the next data arrives. The kind of batch optimisation used in typical reconstruction algorithms is fundamentally unsuited to the real-time domain, since this constant time-step constraint is not obeyed. For this reason, the algorithms developed for real-time SLAM in robotics using predominantly sensors other than vision will be more relevant to the problem of real-time visual localisation and mapping, a point elaborated on in [4].

1.3 The Rest of this Paper

In this paper, in addition to general discussion we will present the basic framework for a real-time single-camera localisation system based on the EKF.

The key points of the approach we propose are:

1. A general model for smooth motion
2. Sparse mapping of a useful selection of high-quality features
3. Active measurement of features selected by information content

2 Representing 3D Position and Orientation

We define the following coordinate frames (see Figure 3):

1. W, the world coordinate frame, defined such that the y axis points directly up and the x and z axes are horizotal.
2. R, the robot frame, fixed with respect to camera. and aligned such that its y axis points to the top of the camera, z to the front and x to the left.

Position and orientation in 3D can be represented minimally with 6 parameters: 3 for position and 3 for orientation. However, we take here the approach of using an extra parameter when representing 3D orientation, and use a **quaternion** which is a way to represent 3D orientation with 4 parameters. Quaternions have the advantages of mathematical convenience and a lack of singularities as a representation for 3D orientation.

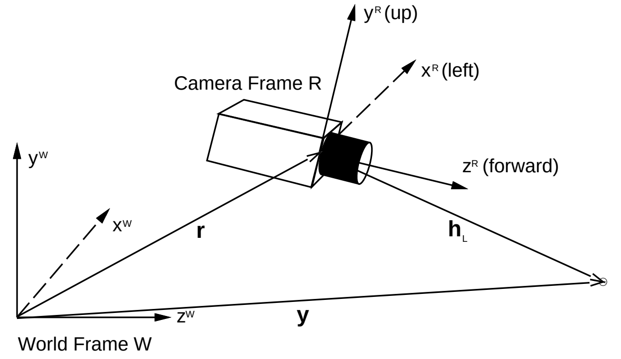


Figure 3: Coordinate frames and vectors in camera geometry: fixed world frame W and robot frame R carried by the camera. The vectors involved in measurement of a feature: robot position $\mathbf{r}$, cartesian measurement $\mathbf{h}_L$ and feature position $\mathbf{y}$ satisfy $\mathbf{y} = \mathbf{r} + \mathbf{h}_L$.

The vector of 7 parameters chosen represent position and orientation is therefore:

$$\mathbf{x}_p = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} \tag{1}$$

We refer to $\mathbf{x}_p$ as the **position state** of a robot or body: a standard way to define 3D position and orientation which is common for any type of robot. We differentiate between $\mathbf{x}_p$ and $\mathbf{x}_v$, the actual **state** of a robot, which may include parameters additional to those representing pure position — these extra parameters may represent parts of a robot which move redundantly with respect to overall position, or other aspects of interest: in the motion model we shall present later, we store estimates of the camera's velocity and angular velocity as well as of position.

The quaternion $\mathbf{q}^{WR}$ is uniquely associated with the rotation matrix $\mathtt{R}^{WR}$ defining the transformation between frames R and W. If $\mathbf{r}^W$ is zero and $\mathtt{R}^{WR}$ is identity, frames W and R coincide (the robot is at the origin of coordinates of the world frame).

3 A Motion Model for a Generally Moving Camera

Constructing a motion model for an agile camera which may for instance be attached to a person's head 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 we were 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; without imposing strong domain constraints, the best we can hope to do is make a model along the lines of only permitting motions with certain maximum accelerations and therefore smoothness. 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 statistical 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's head, it takes account of the unknown intentions of the person, but these too can be statistically modelled.

An intermediate step between the two cases we have discussed is that of a wheeled robot moving on a non-planar surface whose undulations are not known in advance, as studied in [3] in perhaps the first work on SLAM in full 3D. Here, a model for robot motion on a locally flat surface was combined with a model for unknown surface shape which depended on just one parameter (standard deviation of curvature). The result was that uncertainty in position, and particularly in orientation, increased much more rapidly than in the planar case due to the lack of knowledge about surface orientation.

In the case of our agile camera, the type of model we will use 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. (Note that a more sensible model in many circumstances where motion occurs within a bounded area may be an auto-regressive model, where statistically we expect a rapidly moving object to slow down or change direction rather than increase it's speed.) The implication of this model is that we are imposing a certain smoothness on the camera motion: very large accelerations are relatively unlikely.

The fact that we directly model the velocity of the camera in this way means that we must augment the robot position state vector $\mathbf{x}_p$ with velocity terms to form the robot state vector:

$$\mathbf{x}_v = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \\ \mathbf{v}^W \\ \omega^W \end{pmatrix} . \tag{2}$$

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 robot 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, an unknown impulse of acceleration and angular acceleration

$$\mathbf{n} = \begin{pmatrix} \mathbf{V}^W \\ \mathbf{\Omega}^W \end{pmatrix} \tag{3}$$

is applied to the robot. 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:

$$\begin{aligned} \mathbf{f}_v &= \begin{pmatrix} \mathbf{r}^W_{new} \\ \mathbf{q}^{WR}_{new} \\ \mathbf{v}^W_{new} \\ \omega^W_{new} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{r}^W + (\mathbf{v}^W + \mathbf{V}^W)\Delta t \\ \mathbf{q}^{WR} \times \mathbf{q}((\omega^W + \mathbf{\Omega}^W)\Delta t) \\ \mathbf{v}^W + \mathbf{V}^W \\ \omega^W + \mathbf{\Omega}^W \end{pmatrix} . \end{aligned} \tag{4}$$

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 robot after this motion. We find $\mathtt{Q}_v$ via the Jacobian calculation:

$$\mathtt{Q}_v = \frac{\partial \mathbf{f}_v}{\partial \mathbf{n}} \mathtt{P}_n \frac{\partial \mathbf{f}_v}{\partial \mathbf{n}} \ , \tag{5}$$

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 would increase 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.

4 Incorporating Visual Feature Measurements

The features used as landmarks in a visual SLAM system are image interest regions detected with a saliency operator [11] and matched using image correlation (see Figure 4, or [2] for much more detail). We use image patches which are larger (around 15 × 15 pixels) than those typically used in structure from motion, since the features must be highly distinguishable to act as stable long-term landmarks rather than behave as transient tracking points.

Considering the vector sum of Figure 3, the position of a point feature relative to the robot is given by:

$$\mathbf{h}_L^R = \mathtt{R}^{RW}(\mathbf{y}^W - \mathbf{r}^W) \ . \tag{6}$$

Here $\mathbf{h}_L^R$ is the cartesian vector from the sensor centre to the feature. A given sensor will not directly measure the cartesian vector, but some vector $\mathbf{h}$ of parameters which is a function of $\mathbf{h}_L^R$:

$$\mathbf{h} = \mathbf{h}(\mathbf{h}_L^R) \tag{7}$$

In the particular case of making measurements with a single camera, the measurements achieved from the observation of a feature are its horizontal and vertical image positions (u, v). Figure 5 shows the pinhole camera model used:

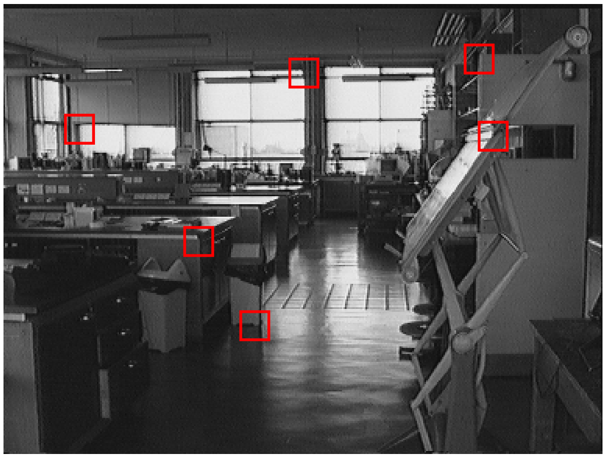


Figure 4: Typical features used in a visual mapping system: an image interest operator locates patches with high intensity gradients in both the x and y directions. These typically correspond to the corners of scene objects and are well localised in space, though reflections or depth discontinuities can also throw up unsuitable candidates: in SLAM, these can rejected over time since they do not behave as stationary landmarks when observed from new viewpoints.

$$\mathbf{h} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - fk_u \frac{h_x^R}{h_z^R} \\ v_0 - fk_v \frac{h_y^R}{h_z^R} \end{pmatrix} \ . \tag{8}$$

Parameters k_u and k_v are the pixel element densities (in pixels per metre) in the u and v directions respectively. The noise covariance $\mathtt{R}$ of this measurement is taken to be diagonal with magnitude determined by image resolution.

A clear characteristic of this measurement model is that it is **not invertible**: that is to say that while it tells us the value of an image measurement given the position of the camera and a feature, it cannot be inverted to give the position of a feature given image measurement and camera position. This is obvious once we consider the projective character of visual measurement: the depth of scene features is lost. 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. From just one view, all that can be initialised into the map is a ray in space on which it is known that the feature must lie. At least one other view of the feature from a different camera position must then be obtained so

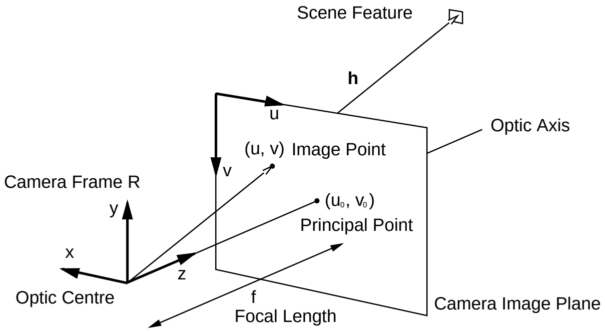


Figure 5: The standard pinhole camera model with focal length f and principal point (u_0, v_0). For ease of understanding, the figure shows the image plane in front of the optic centre, though of course in a real camera it is behind and inverted.

that the features's 3D position can be estimated. This is a task which is not currently solved in our implementation. There are important issues with respect to search to be solved: when the camera has reached a new position and wished to obtain a second view of a partially-initialised feature, where should it search in the image to find it? Clearly along the "epipolar line" which is the image of the ray the feature must lie on in the current view, but the details of search regions present an involved problem when all the coupled uncertainty in the system is taken into account.

5 An Experiment in Real-Time Single Camera Localisation

Since one of the outstanding problems of single-camera SLAM is feature initialisation, as a first step an experiment was carried out in which feature positions were measured by hand and initialised into a map as known features (see Figure 6: the features were corners of some of the squares on a calibration grid and a piece of paper on the floor). A total of 12 features were mapped.

Starting from rest in a known position, a hand-held camera was waved in front of the scene during 6 seconds and images were captured at 30 frames per second. In an EKF implementation (using a full-covariance map for the 12 features despite the fact that in this particular experiment their perfectly-known positions were in fact uncoupled), at each frame first a predictive update was performed based on the constant-velocity motion model, then a measurement update based on a measurement of just **one** feature per frame.

An **active** measurement strategy was used, similar to that used in our previous work using movable cameras [2, 5, 3]. An information criterion made choices about which feature to measure at each time step with the essential result that the measurement was chosen **whose result was least predictable**. The effect of this is to keep estimates consistently good by continuously locking down the largest uncertainty available in the system. In practice, the criterion recommends very rapid switching of attention between different features. As opposed to our previous work with movable cameras, where a penalty was associated with changing fixation from one feature to the other, the purely digital fixation switching occuring here is costless and can be undertaken freely.

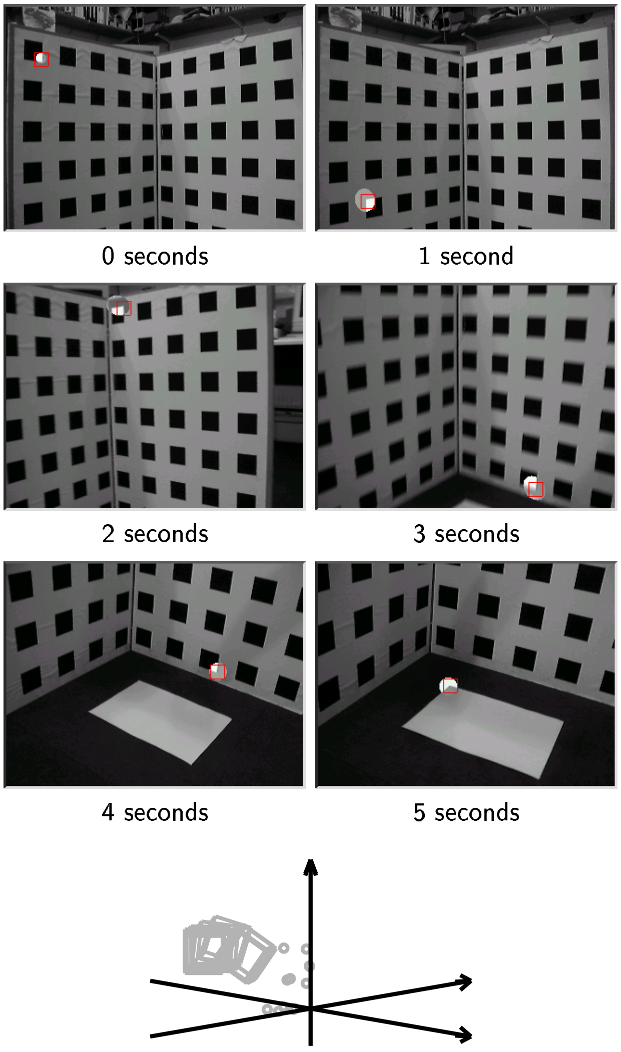
Image and map processing were carried out on a standard laptop PC and the required processing per frame was achieved in a small amount less than the $\frac{1}{30}s$ available in a real-time budget.

Camera motion estimation proceeded accurately for around 6 seconds, when a lack of visible features led to a rapid growth in uncertainty, large feature search regions and eventually a mismatch (failure of data association: the search window was so big that more than one part of the grid with similar appearance was included) which caused accurate estimation to fail. Although we do not have ground-truth data for the motion, the fact that accurate tracking of all the features was maintained during the 6 seconds of successful estimation: this is a very promising result indeed, based as it was on measurement of just **a single feature** at each time step, and a real validation of the active measurement strategy, and a sharp contrast with structure from motion systems where dozens or hundreds of different features are measured in each frame: the message is clearly that the information gained by such dense measurement is highly redundant and it is far better to concentrate on a few, high quality features if camera localisation is the goal.

Increased robustness and accuracy will be achieved using more feature measurements per step, but at a computational cost. The factor of robustness is key: if we wish to have a system which does not fail with one mismatched feature, immediate verification of each proposed match must be available from attempted matches of other features; if a consistent set of matches is found, using a technique such as RANSAC, incorrect matches can be pruned out.

6 Conclusions

We have introduced the issues involved in tackling SLAM with a single camera and presented a preliminary implementation to perform map-based localisa-



0 seconds

1 second

2 seconds

3 seconds

4 seconds

5 seconds

Scene points and recovered 3D camera trajectory

Figure 6: Experiment in real-time camera localisation using a known map of features. Images were received and processed at 30 frames per second on a standard 400MHz laptop PC. Only one feature measurement was made in each frame, the feature to be measured chosen based on visibility and information content criteria. Ellipses in the images show 3σ search regions: these were the only image regions needing to be processed in each frame.

tion in real-time with the very promising result that we can get good results with just one feature measurement per frame. A full implementation of the demonstration described in this paper is available open-source and ready-to-run for Linux as part of the “Scene” C++ software library [4] for sequential localisation and map-building at:

http://www.robots.ox.ac.uk/~ajd/Scene/

The research issues which we will focus on in the near future are as follows:

1. Feature initialisation from multiple views: features must be viewed from two significantly different viewpoints before their 3D positions can be initialised, and care must be taken that they are inserted into the map with the correct uncertainty.

2. Multiple hypotheses and non-Gaussian probability distributions: while the EKF has often been shown to perform well in SLAM, there will be many cases with the sparse measurements of single camera SLAM where it is desirable to propagate multiple hypotheses over time for later resolution. Since generalised schemes for representing non-Gaussian probability densities suffer from scaling problems, explicit schemes for multiple Gaussian hypotheses may suffice.

3. Pure information theoretic searching: 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?

4. Local sensors such as accelerometers and gyros may be permissible in some applications and are expected to a have large positive effect, dramatically reducing visual search regions.

5. Map scaling: in this as in all SLAM problems, the problem of computational cost in large maps arises, and methods such as the postponement of map updates [2, 8] will be implemented.

6. Different camera/lens types: it is expected that cameras with a wide field of view will be especially useful for localisation despite their low angular resolution, since they permit smaller sets of features to be visible through large motions.

References

[1] K. S. Chong and L. Kleeman. Feature-based mapping in real, large scale environments using an ultrasonic array. *International Journal of Robotics Research*, 18(2):3–19, January 1999.

[2] A. J. Davison. *Mobile Robot Navigation Using Active Vision.* PhD thesis, University of Oxford, 1998. Available at http://www.robots.ox.ac.uk/˜ajd/.

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

[4] A. J. Davison and N. Kita. Sequential localisation and map-building for real-time computer vision and robotics. *Robotics and Autonomous Systems*, 36(4):171–183, 2001.

[5] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. Accepted for publication in IEEE PAMI, to appear, 2002.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the 6th International Conference on Computer Vision, Bombay*, pages 90–96, 1998.

[11] 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.

[12] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31, 1998.