

3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain

Andrew J. Davison
Robotics Research Group
Department of Engineering Science
University of Oxford, UK
ajd@robots.ox.ac.uk

Nobuyuki Kita
Intelligent Systems Institute
AIST
Tsukuba, Ibaraki 305-8568, Japan
n.kita@aist.go.jp

Abstract

Work in simultaneous localisation and map-building (“SLAM”) for mobile robots has focused on the simplified case in which a robot is considered to move in two dimensions on a ground plane. While this is often a good approximation, a large number of real-world applications require robots to move around terrain which has significant slopes and undulations, and it is desirable that these robots too should be able to estimate their locations by building maps of natural features. In this paper we describe a real-time EKF-based SLAM system permitting unconstrained 3D localisation, and in particular develop models for the motion of a wheeled robot in the presence of unknown slope variations. In a fully automatic implementation, our robot observes visual point features using fixating stereo vision and builds a sparse map on-the-fly. Combining this visual measurement with information from odometry and a roll/pitch accelerometer sensor, the robot performs accurate, repeatable localisation while traversing an undulating course.

1. Introduction

The vast majority of research into simultaneous robot localisation and map-building (SLAM) has considered the case where the robot under consideration moves on a flat ground-plane. While this is a reasonable assumption to make in many situations, such as for wheeled robots moving in certain man-made areas, it is severely limiting when we consider other areas where mobile robots could potentially be deployed, in outdoor or more challenging indoor environments — or in the case of advanced robots such as humanoids which have more general types of motion. These applications clearly call for robots which can estimate their locations and build maps in three dimensions.

While the number of parameters needed to represent cartesian position in 3D simply increases by one in the move from 2D, representing orientation in 3D requires minimally 3 parameters, compared with the 1 needed for 2D orienta-

tion. The tangled nature of these orientation parameters is what makes 3D localisation mathematically involved.

In this paper, we present a detailed framework for estimating the 3D motion of a robot in the case that it is a wheeled vehicle moving over a non-flat surface whose changes in slope are unknown. This will be the case in most applications involving uneven ground because generally a robot will not have a map of the terrain in advance. We will refer specifically to the Nomad 200 robot we have used in experiments where necessary, though most of the discussion applies to any wheeled robot. We present models for the incorporation of information from sensors able to make measurements of arbitrary features in the surroundings, focusing our discussion on active vision. We show that these visual measurements can be combined with readings from robot odometry and an accelerometer sensor in a rigorous fashion to produce estimates which benefit from all available information sources.

Structure from motion in computer vision and simultaneous map building and localisation for mobile robots are two views of the same problem: estimation of the motion of a body which moves through a static environment about which it has little or no prior knowledge, using measurements from its sensors to provide information about its motion and the structure of the world. This body could be a single camera, a robot with various sensors, or any other moving object able to sense its surroundings. Nevertheless, in recent years research has taken many paths to solving this problem with a lack of acknowledgement of its general nature, a particular divide arising between robotic and “pure vision” approaches.

Crucially, in this paper we are interested in the **sequential** case, where map-building and localisation are able to proceed in a step-by-step fashion as movement occurs. This contrasts with situations where the **batch** methods currently prevalent in computer vision [6] can be applied, in which measurements from different time steps are used in parallel after the event. Despite renewed interest in sequential map-

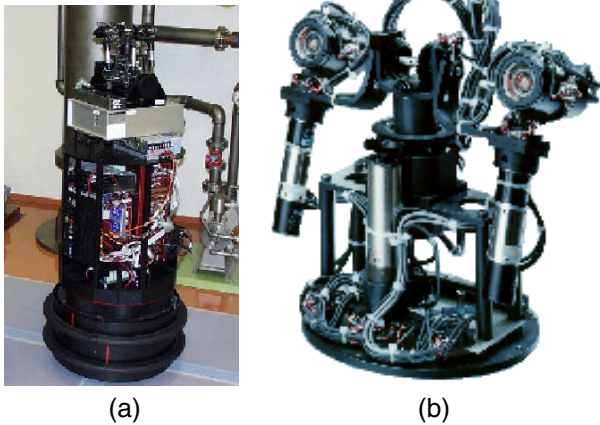


Figure 1: (a) Robot carrying active stereo head (b), featuring 4 degrees of rotational freedom and foveated lenses [8].

building in mobile robotics, in computer vision recent successful work on reconstruction from image sequences has conspicuously not been accompanied by advances in real-time methods. In the mobile robotics literature, sequential localisation and map-building is now increasingly well understood, although current successful implementations are based almost exclusively not on vision but sonar or laser range-finder sensors [11, 2, 5, 9], and build maps in 2D. In particular, the critical role of maintaining information about the coupling between estimates has been proven, highlighting the deficiencies in early approaches, amongst which vision played a significant part [7], which were unable to build persistent maps of long-term use due to “motion drift”. Easing the computational burden of SLAM while maintaining this cross-correlation information as maps get bigger is the current focus of SLAM research.

With this paper, we hope to place another stepping stone between the areas of computer vision and robotics, and draw attention to the fact that sequential localisation and map-building in 3D is exactly the problem which when solved will lead to new wealth of real-time vision applications, such as real-time camera position tracking for virtual studios or virtual reality applications, as well as opening up new applications domains in robotics. After introducing and reviewing active-vision based localisation and map-building in Section 2, closely following the approach of [4, 3], we look at representing 3D robot position in Section 3, and construct models for motion on unknown, uneven terrain in Section 4. In Sections 5 and 6 we explain how measurements from active vision and a roll/pitch sensor respectively can be incorporated into a 3D estimation framework. Experimental results demonstrating repeatable localisation over undulating terrain are presented in simulation in Section 7 and in robot implementation in Section 8.

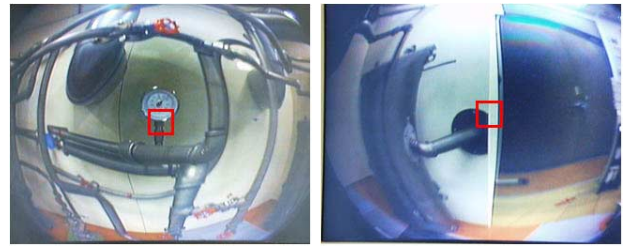


Figure 2: Fixated views of typical feature matches by correlation of 15×15 pixel image patches.

2 SLAM Using Active Vision

2.1 Active Vision

In active approaches to sensing, sensor or information processing resources are directed purposively to regions of current interest in a scene, rather than being used to acquire and process data uniformly. In vision, active operation is achieved either by selective processing of the images acquired by fixed cameras, or in our case by physically directing the cameras as required using a motorised camera platform or “active head”: see Figure 1. While active vision is easily applied to short range “tactical” navigation, there have been few attempts to apply it to more long-term navigation tasks such as map-building and using [3].

2.2 Visual Features

The basis for localisation using vision is a map of static features whose positions relative to the robot are repeatedly measurable. Within a general mapping framework, there is the potential for these features to have many different forms: points, lines or planes for instance. In our current implementation, point features in 3D space are recognised using image correlation matching, which proves to be surprisingly robust to changes in viewpoint. The active head moves to fixate features for measurement with both of its cameras, acting as an accurate “pointing stick” which can measure the direction and depth of features over a very wide field of view. Figure 2 shows some typical features used. These features are detected automatically by the robot as it navigates using an image interest operator [10]. Active vision lends itself to building a **sparse** map of **high-quality, widely-spaced** features. This kind of feature map, used with a strategy of selective measurement, provides a robot with the focused information which most aids localisation.

2.3 Storing and Updating Map Data

The current state of the robot and the scene features which are known about is stored in the system state vector \hat{x} and

covariance matrix P . These are partitioned as follows:

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \end{pmatrix}, \quad P = \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & \cdots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & \cdots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (1)$$

$\hat{\mathbf{x}}_v$ is the robot state estimate, and $\hat{\mathbf{y}}_i$ the estimated state of the i th feature ($\hat{x}, \hat{y}, \hat{z}$ for the case of a 3D point). The state vector represents the robot's map of its environment and its place within it, and the covariance matrix how uncertain this information is.

Like most current SLAM implementations, we propagate the covariances between the robot state and all feature estimates, and between the feature states themselves. This is particularly critical in our system, since the small number of features generally used must have estimates which are of very high quality to provide accurate localisation information. The ability of active cameras to view features over a huge field of view is key to the quality of this map: the robot can really see the same features continuously as it goes through very large motions and rotations; thus fewer features need to be added to the map, and the uncertainties related to those present can be reduced successively as the robot is able to measure them repeatably over long periods.

The data is updated sequentially as the robot moves around its environment and makes measurements of the features in its map following the rules of the Extended Kalman Filter: a prediction step when the robot moves, and an update step when a measurement is made of a feature.

2.4 Active Measurement

In an active scenario, it is necessary to decide at each instant which feature in the map to attempt to measure. This decision is made based on two criteria: expected visibility and the information content of the measurement. Expected visibility (more precisely measurability) depends on the details of sensor and feature type; with point features matched by correlation, we do not expect to be successful with matching if the viewpoint is too different from that from which the features were initially seen. Since we have an estimate of the current robot position, the predicted viewing direction can be evaluated in this respect.

Once a measurable subset of features in the map has been identified, the information to be gained by measuring each one is compared: essentially we choose the measurement which has a high innovation covariance [4], the principle being that there is little use in making a measurement of which the result is highly predictable. The selected feature is then measured by driving the active head to the angles predicted for fixation on that feature, and searching the images obtained for a match. Precise search regions are calcu-

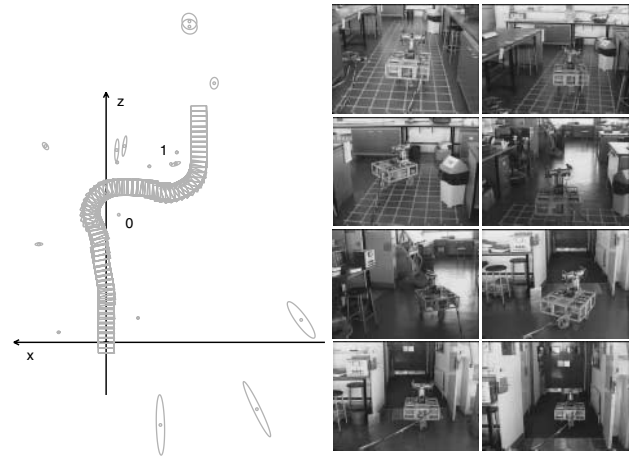


Figure 3: 2D navigation results from [4] in which a robot builds a map of point features of to aid position-based localisation while steering a twisting course.

lated from the uncertainty in the map, which maximise computational efficiency and reduce the chance of mismatches.

2.5 2D Navigation Results

Active-vision based localisation, when implemented with planar robot motion but building a 3D map of features in [4] resulted in a system able to perform repeatable, autonomous navigation using naturally-occurring features: see Figure 3.

3. 3D Position and Orientation

Position and orientation in 3D can be minimally represented with 6 parameters: 3 for position and 3 for orientation. Here we add an extra parameter, and use a **quaternion** to represent 3D orientation with 4 parameters. Quaternions have advantages of mathematical convenience and a lack of singularities as a representation for 3D orientation. Position and orientation is represented by the 7-parameter vector:

$$\mathbf{x}_p = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \end{pmatrix} \quad (2)$$

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 the actual **state** \mathbf{x}_v of a robot, which may include parameters additional to those representing pure position — these extra parameters may represent parts of the robot which move redundantly with respect to overall position, perhaps velocity or acceleration estimates, or other aspects.

In the case of the robot specifically considered in this paper, there is one extra parameter which we choose to insert into the state \mathbf{x}_p : the relative steering angle S , which is the

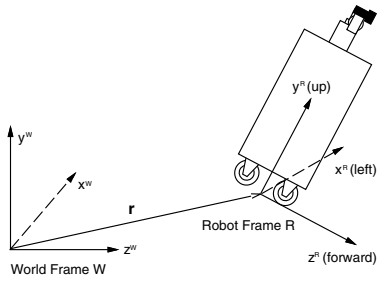


Figure 4: Coordinate frames: fixed world frame W and robot frame R carried by the robot.

angle between the robot’s rotating turret and the direction of its three mechanically coupled wheels. The robot’s state vector becomes:

$$\mathbf{x}_v = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \\ S \end{pmatrix}. \quad (3)$$

We define two coordinate frames: W , the static world coordinate frame, and R , the robot frame, fixed with respect to the robot’s turret. If \mathbf{r}^W is zero and $\mathbf{R}^{WR} = \mathbf{R}(\mathbf{q}^{WR})$ is identity, frames W and R coincide: see Figure 4.

4. Motion on Undulating Terrain

In 2D navigation, given a robot’s current position and velocity and steering control inputs it is possible to calculate an estimate of its motion during a small time interval whose uncertainty depends only on the uncertainty in its odometry readings. However, when moving in 3D on undulating terrain this is not the case: the orientation of the terrain could change by an unknown amount in the interim. Clearly, if a robot is equipped only with such odometry and no other sensors, we will not get far with 3D position estimation because wheel odometry does not give any information about slope. However, before information from other sensors can be incorporated it is crucial to have a sensible model of what **odometry alone** tells us about the robot’s position.

Some assumption must be made about the properties of the surface on which the robot is driving; we propose the following relatively simple model:

- At every point on the surface, the terrain has a curvature about a single axis whose unknown direction is evenly distributed and whose unknown magnitude is normally distributed.

We also implicitly assume that the robot is in full contact with the surface at all times (that it does not tip up or “jump”), and therefore that knowing the robot’s orientation at any point is equivalent to knowing the orientation of the section of surface beneath.

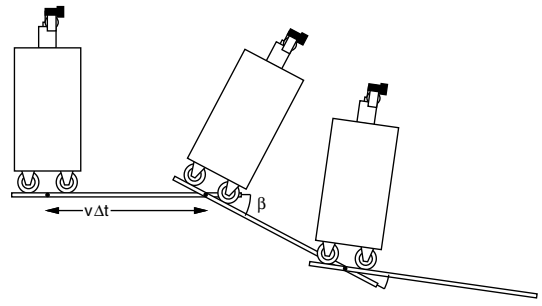


Figure 5: The “piecewise planar” approximation to an undulating surface, shown here with exaggerated step size and a simplified side-on cut-through of the real 3D situation.

This surface model requires the specification of only one parameter: the standard deviation of the curvature σ_C , which is assumed to be of zero mean and of Gaussian distribution. This parameter reflects how flat (low value) or undulating (high value) the surface on which the robot will run is expected to be.

In previous work on 2D localisation [3], we have been able to use motion models which are valid for any size of time increment Δt and did not rely on time steps being short. However, in the 3D case, mathematical complication causes us currently to abandon the possibility of building a smooth 3D model of this type, where for instance the robot was assumed to follow sections of surface of constant curvature. We therefore make a “piecewise planar” approximation to the undulating surface on which the robot drives: as shown in Figure 5, we assume that each $v\Delta t$ displacement of the robot occurs along a locally planar surface, and is followed by an instantaneous change in orientation through angle β when a “fold” in the surface is encountered. While Figure 5 shows a simplified side-on view, in full 3D the orientation of the axis about which this orientation change occurs is assumed to be evenly distributed within the plane on which the robot is currently moving: that is to say that as well as slope changes which will cause it to rock forward and back, there will also be those that cause it to rock from side to side or about any axis in between.

The motion of the robot can therefore be broken down into two steps:

1. A planar motion, of size derived from velocity and steering control parameters, within the plane defined by the robot’s current position and orientation.
2. An unknown pure rotation as the robot crosses to a new planar patch.

In a sequential estimation framework, we must calculate a new state estimate $\mathbf{f}_v(\mathbf{x}_v, \mathbf{u})$ and the increase in state uncertainty (process noise covariance) \mathbf{Q}_v for the robot after

this motion.

In step 1, the planar motion, the robot moves in the xz plane of its local frame R (this is the local “ground plane”). The details of this motion will depend on the robot’s control system. For our holonomic robot, we assume that the planar motion itself is made of two parts: a pure rotation, during which both the turret and steering wheels rotate, and a pure translation along the direction defined by the new direction of the steering wheels (we would need to combine these two motions into one for a car-like, non-holonomic robot). This model describes the case that the robot is driven in a stop-start fashion, only steering when stationary, but is a good approximation in continuous motion provided that the rotations happen quickly compared to the speed of translation.

After the “turn” step, the new state of the robot becomes:

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r}_{new}^W \\ \mathbf{q}_{new}^{WR} \\ S_{new} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{q}^{WR} \times \mathbf{q}(\Delta T) \\ S + \Delta S - \Delta T \end{pmatrix}, \quad (4)$$

and after the translation step,

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r} + \mathbf{R}(\mathbf{q}^{WR} \times \mathbf{q}(S + \Delta S)) \\ \mathbf{q}^{WR} \times \mathbf{q}(\Delta T) \\ S + \Delta S - \Delta T \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ v\Delta t \end{pmatrix}. \quad (5)$$

Here ΔT is the increment in turret angle, and ΔS the steering increment. $\mathbf{q}(\Delta T)$ and $\mathbf{q}(S + \Delta S)$ are the quaternions which represent the rotations in the robot frame associated with ΔT and $S + \Delta S$; since these rotations are purely about the y axis of the robot frame they are trivially formed:

$$\mathbf{q}(\Delta T) = \begin{pmatrix} \cos \frac{\Delta T}{2} \\ 0 \\ \sin \frac{\Delta T}{2} \\ 0 \end{pmatrix}, \quad \mathbf{q}(S + \Delta S) = \begin{pmatrix} \cos \frac{S + \Delta S}{2} \\ 0 \\ \sin \frac{S + \Delta S}{2} \\ 0 \end{pmatrix}. \quad (6)$$

The notation $R(\mathbf{q})$ represents the rotation matrix uniquely defined by quaternion \mathbf{q} .

Turning now to the unknown slope change part of our motion model, the size of the random slope change β can be derived from our assumption about the undulations of the surface. If the standard deviation of the surface curvature is σ_C , then the standard deviation of β is:

$$\sigma_\beta = v\Delta t\sigma_C. \quad (7)$$

(Since curvature is defined as change of angle per unit distance along a curve, we multiply σ_C by $v\Delta t$ to obtain an absolute angle change.)

The effect of the random slope change is to induce an additional pure rotation of the robot. The quaternion defining

this rotation we call \mathbf{q}_s . We now have a complete expression for the change of state of the robot in time δt :

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r} + \mathbf{R}(\mathbf{q}^{WR} \times \mathbf{q}(S + \Delta S)) \\ \mathbf{q}^{WR} \times \mathbf{q}(\Delta T) \times \mathbf{q}_s \\ S + \Delta S - \Delta T \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ v\Delta t \end{pmatrix}. \quad (8)$$

Note that here \mathbf{q}_s , representing the unknown slope rotation, is a random variable with **zero mean effect**. This means that we can ignore it when calculating the value of the new state estimate, and simply use the expression in Equation 5 above. However, its role becomes clear when we calculate \mathbf{Q}_v , the process noise of the movement, or increase in state uncertainty it induces.

We break the calculation of \mathbf{Q}_v into two steps. First, for the planar part of the motion, we assume that uncertainty in the state update will be due to uncertainty in the value of the control parameters. The control vector \mathbf{u} , containing parameters v , ΔS and ΔT , has a diagonal covariance matrix since we assume the parameters to be uncorrelated:

$$\mathbf{u} = \begin{pmatrix} v \\ \Delta S \\ \Delta T \end{pmatrix}, \quad \mathbf{P}_u = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & \sigma_{\Delta S}^2 & 0 \\ 0 & 0 & \sigma_{\Delta T}^2 \end{bmatrix}, \quad (9)$$

and thus the process noise for the planar part of the motion can be calculated as:

$$\mathbf{Q}_v(1) = \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}} \mathbf{P}_u \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}}^\top, \quad (10)$$

We do not go into detail here about calculating the Jacobian $\frac{\partial \mathbf{f}_v}{\partial \mathbf{u}}$ except to say that it is involved but tractable: alternatively, there is of course the possibility in implementation to use a numerical approximation to Jacobians, or to form them using the symbolic differentiation capabilities of mathematical software packages.

To now modify $\mathbf{Q}_v(1)$ to include the effect of slope-induced rotation, we must form the covariance matrix of \mathbf{q}_s , the random quaternion. We first parameterise \mathbf{q}_s in terms of two angles: β , the Gaussian magnitude of the rotation to which we have already referred, and ϕ , the orientation of the axis, lying in the robot frame’s xz plane, about which the rotation occurs: ϕ is evenly distributed in $0 \rightarrow 2\pi$.

$$\mathbf{q}_s = \begin{pmatrix} \cos \frac{\beta}{2} \\ \cos \phi \sin \frac{\beta}{2} \\ 0 \\ \sin \phi \sin \frac{\beta}{2} \end{pmatrix} \quad (11)$$

An analysis of this vector reveals its covariance matrix:

$$\mathbf{P}_{q_s} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\sigma_\beta^2}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sigma_\beta^2}{8} \end{bmatrix} \quad (12)$$

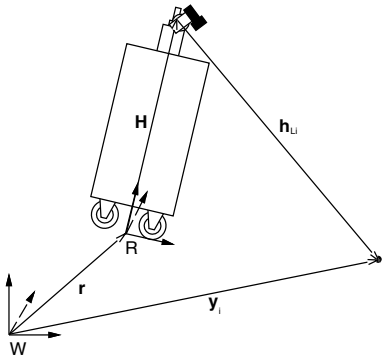


Figure 6: Vectors involved in measurement of a feature: robot position \mathbf{r} , sensor offset \mathbf{H} , cartesian measurement \mathbf{h}_{Li} and feature position \mathbf{y}_i satisfy $\mathbf{y}_i = \mathbf{r} + \mathbf{H} + \mathbf{h}_{Li}$.

(where $\sigma_\beta = v\Delta t\sigma_C$ as shown above).

The final value of Q_v is calculated with the equation:

$$Q_v = Q_v(1) + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \mathbf{q}^{WR}(2)}{\partial \mathbf{q}_s} \mathbf{p}_{q_s} \frac{\partial \mathbf{q}^{WR}(2)}{\partial \mathbf{q}_s}^\top & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (13)$$

where $\mathbf{q}^{WR}(2)$ is defined as $\mathbf{q}^{WR}(2) = \mathbf{q}^{WR}(1) \times \mathbf{q}_s$, with $\mathbf{q}^{WR}(1)$ the value of \mathbf{q}^{WR} after the planar part of the motion update. Note that for implementation in a Kalman Filter, Equation 5 must also be differentiated to provide $\frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}$.

It is interesting to consider the way in which we will use this model (which incidentally we acknowledge potentially to be topologically naive): only local use will be made of it — that is to say that as the robot moves over each part of the terrain, the model will be used to calculate how the uncertainty in robot position and orientation should be increased due to lack of knowledge about the surface orientation. Presumably, other sensors will then be used to give more information about these unknown quantities. However, in this work we will not attempt to save this information and map the surface undulations to feed back into the model later. If we pass over the same region again, the model will be used to increase the uncertainty by the same amount. We will rely on measurements from other sensors of stable features to provide repeatable robot position estimates.

5. Incorporating Feature Measurements from Active Vision

We now consider incorporating information from measurement of features external to the robot. In the current discussion we will consider only the case of point features: that is to say features which have well defined point positions in 3D space with no significant volume.

Considering the vector sum of Figure 6, the position of a point feature relative to the robot is given by:

$$\mathbf{h}_{Li}^R = \mathbf{R}^{RW}(\mathbf{y}_i^W - (\mathbf{r}^W + \mathbf{H}^W)). \quad (14)$$

Here \mathbf{h}_{Li} is the cartesian vector from the sensor centre to the feature, and \mathbf{H} is the offset from the robot's centre of coordinates to the centre of the sensor.

A given sensor will not directly measure the cartesian vector to a feature, but some vector \mathbf{h}_i of parameters which is a function of this vector: $\mathbf{h}_i = \mathbf{h}_i(\mathbf{h}_{Li})$. In the particular case of making measurements with an active head, from image measurements we calculate idealised pan, elevation and vergence angles α_i , e_i and γ_i which have the following functional relationship to \mathbf{h}_{Li} :

$$\mathbf{h}_i = \begin{pmatrix} \alpha_i \\ e_i \\ \gamma_i \end{pmatrix} = \begin{pmatrix} \tan^{-1} \frac{h_{Lix}}{h_{Liz}} \\ \tan^{-1} \frac{h_{Liy}}{h_{Liz}} \\ \tan^{-1} \frac{1}{2h_{Li}} \end{pmatrix}. \quad (15)$$

The covariance \mathbf{R}_i of this measurement is taken diagonal with magnitude determined by image resolution [3].

6. Incorporating Measurements from a Roll/Pitch Sensor

A roll/pitch sensor mounted on a robot measures its deviation from horizontal via an accelerometer mechanism: assuming that the robot is moving at a uniform velocity (or stationary), the sensor detects the direction of the force of gravity and therefore knows which way is “down”. Of course a measurement from this sensor is not sufficient to deduce the overall 3D orientation of the robot: it provides no information about its orientation about the axis perpendicular to the down direction (in the aviation-style terminology from which the terms roll and pitch come, this third angle is referred to as “yaw”). Nevertheless, regular measurements from a roll/pitch sensor can aid the localisation of a robot moving on undulating terrain to a large degree, reducing visual search regions.

The angles measured by our roll/pitch sensor, which is fixed rigidly to the robot, are:

$$\mathbf{h}_v = \begin{pmatrix} \theta_R \\ \theta_P \end{pmatrix} = \begin{pmatrix} \tan^{-1} \left(\frac{y_{Wx}^R}{y_{Wy}^R} \right) \\ \tan^{-1} \left(-\frac{y_{Wz}^R}{y_{Wy}^R} \right) \end{pmatrix}, \quad (16)$$

where y_{Wx}^R , y_{Wy}^R and y_{Wz}^R are components of the vector \mathbf{y}_W^R , which is the unit vector with the orientation of the world coordinate frame W 's y axis as seen from the robot frame R , defined as:

$$\mathbf{y}_W^R = \mathbf{R}(\bar{\mathbf{q}}^{WR}) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \quad (17)$$

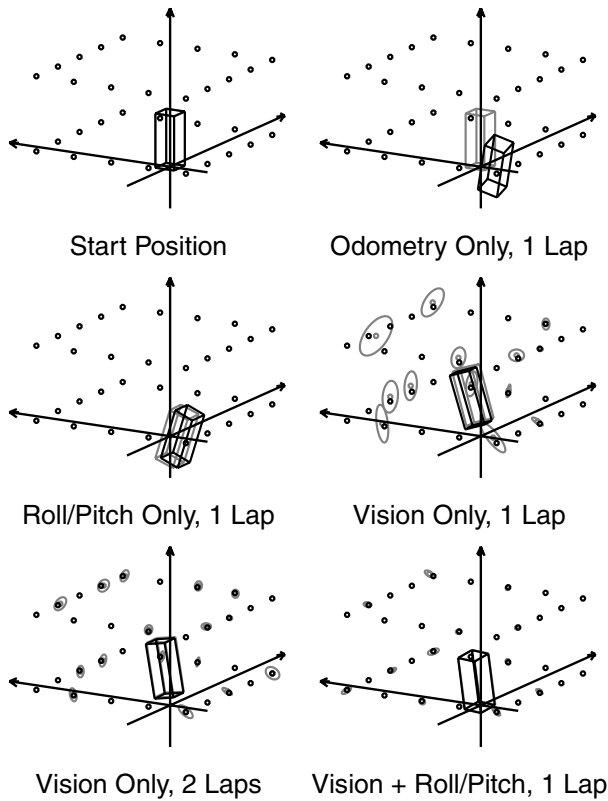


Figure 7: Simulation experiments: grey estimated robot and feature positions are superimposed on black ground truth. 3σ covariance ellipses represent feature uncertainty.

The measurement noise covariance R_v of this measurement is characteristically diagonal and small (a standard deviation of somewhat less than 1° can be expected).

Since there is little implementational difficulty with taking high-frequency measurements from a roll/pitch sensor, it is feasible to make a measurement after every motion step of the robot — this can of course be without stopping the robot’s continuous motion. In this way, we are able immediately to correct the large uncertainty in robot orientation introduced by unknown slope changes.

7. Simulation Experiments

To test the validity of the approach described, simulation experiments were carried out in which stochastic trajectories were generated for the robot by driving its motion model with Gaussian noise: effectively this simulates an undulating surface and the robot’s noisy motion over it. The robot was driven in a loop around an approximately square trajectory of side-length 2m in the vicinity of a regular distribution of feature points of which it was able to make simulated, noisy visual measurements.

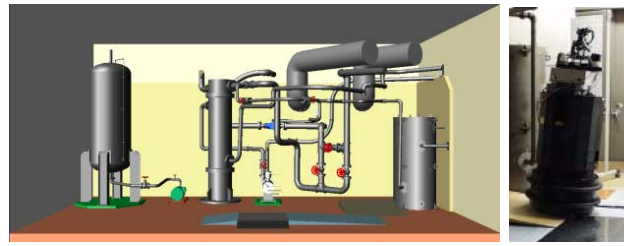


Figure 8: VRML model of the experimental area, showing the slopes installed, and the real robot ascending a ramp.

Results are given in Figure 7, showing the true and estimated positions of the robot and any features measured after laps of this course from the start position using different combinations of sensors. Using no sensors but forming estimates solely from the integration of odometry, after just one circuit the accumulation of odometry errors and the effect of unknown slope changes meant that the robot’s estimated position had diverged significantly from ground truth. Using the roll/pitch sensor dramatically reduced the estimation error due to the information provided on slope changes, but some drift was still evident.

Using vision to build a sparse map of features to which repeated reference could be made provided the means to recover from this drift. However, when vision only was used the map generated after just one circuit included a large degree of uncertainty, as can be seen by the large feature ellipses; the lack of absolute orientation information means that large robot uncertainty becomes coupled with large feature uncertainty. One more lap of the course and further measurements improved the map significantly.

As expected, the best results were achieved when vision and the roll/pitch sensor were used in tandem: in this case an extremely accurate map was generated after just one lap.

8. Robot Experiments

Our ability to conduct challenging experiments was hampered by the fact that our current robot is not well suited to navigating on undulating terrain, being tall, narrow and top-heavy. Nevertheless, ramps with elevation of 10cm and angle 10° of were installed on the floor of the experimental environment (a mockup of an industrial plant) depicted graphically in Figure 8.

The robot was commanded to perform position-based navigation along a route of waypoints taking it over the ramps on an L-shaped journey (moving from left-to-right in Figure 8), passing at one point through a narrow gap between pipes. The robot then returned along the same route to its starting position before retracing its steps. Automatic detection and measurement of scene features was carried out continuously in combination with roll/pitch measure-

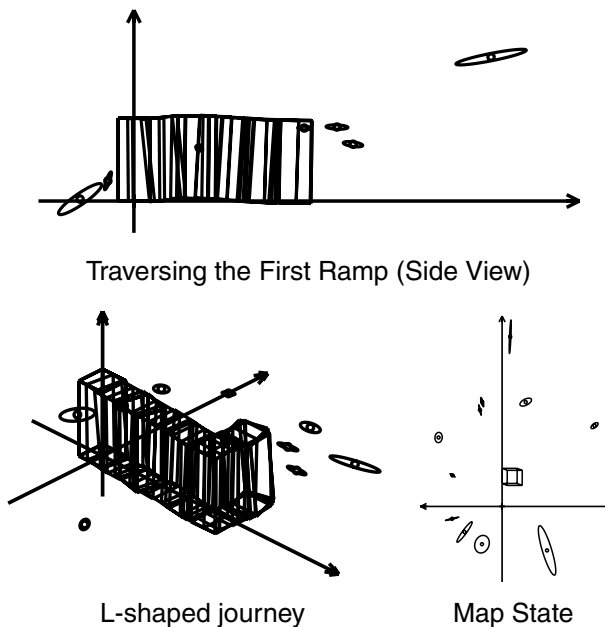


Figure 9: Estimated robot position sequences obtained while traversing the course of ramps.

ments. Vision and map processing was carried out by the robot's onboard Linux PC. Sequences of estimated robot positions shown against the map of features generated are shown in Figure 9, together with the state of the map seen from overhead when the robot had returned to close to its starting position. At this point, after its undulating journey, the robot's position estimate was within 15cm longitudinally of the hand-measured ground-truth location, and within 1.5cm vertically. Most features in the map have been measured many times and have small positional uncertainty.

9. Conclusions

We have given a thorough theoretical exposition and presented experimental implementation of simultaneous localisation and map-building in 3D using active vision. Localisation of this type could extend the operating domain of robots from the flat surfaces of man-made areas to areas of application outdoors or in interplanetary exploration.

We also consider work of this type on sequential visual map-building to be relevant to anyone with an interest in real-time vision, particularly in applications requiring real-time structure from motion systems, the first examples of which have started to appear [1].

A full software implementation of the system described in this paper is available as part of the generalised open-source library "Scene" for simultaneous localisation and map-building from <http://www.robots.ox.ac.uk/~ajd/Scene/>.

References

- [1] 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*.
- [2] 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.
- [3] A. J. Davison. *Mobile Robot Navigation Using Active Vision*. PhD thesis, University of Oxford, 1998. Available at <http://www.robots.ox.ac.uk/~ajd/>.
- [4] A. J. Davison and D. W. Murray. Mobile robot localisation using active vision. In *Proceedings of the 5th European Conference on Computer Vision, Freiburg*, pages 809–825, 1998.
- [5] 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.
- [6] 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.
- [7] C. G. Harris. Geometry from visual motion. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, Cambridge, MA, 1992.
- [8] Y. Kuniyoshi, N. Kita, S. Rougeaux, and T. Suehiro. Active stereo vision system with foveated wide angle lenses. In *Proceedings of the Asian Conference on Computer Vision*, pages 359–363, 1995.
- [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] 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.
- [11] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31, 1998.