ELSEVIER

Robotics and Autonomous Systems 33 (2000) 43–58

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Feature extraction and tracking for scanning range sensors

N.E. Pears

Department of Computer Science, York University, Heslington, York Y010 5DD, UK

Received 26 May 1999; received in revised form 14 February 2000
Communicated by F.C.A. Groen

Abstract

A solution to the problem of simultaneously extracting and tracking a piecewise-linear range representation of a mobile robot's local environment is presented. The classical framework of the extended Kalman filter fits this problem extremely well and the algorithm presented is immune to vehicle motion and active sensor reorientation during the finite capture time of the range scan. The paper also tackles the problems of fast, non-iterative, initialisation of feature tracks, and feature track management over multiple scans. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Range sensing; Feature extraction; Feature tracking; Mobile robot navigation; Laser scanners

1. Introduction

Several different methods and numerous different configurations of range sensing have been investigated for 3D inspection and Robotics applications [3]. These include optical radar [10], sonar [6], projected stripe [5], projected spot [9] and projected pattern [8] schemes. The latter three are based on optical triangulation. This provides an effective solution for short to medium range sensing and systems can readily be reconfigured according to application. Indeed, a recent system has employed variable geometry to extract range maps with different qualities which subsequently may be fused into a single high quality map [18].

We have developed a scanning range sensor, based on optical triangulation to guide obstacle avoidance and docking manoeuvres of a mobile robot. Scanning a collimated laser beam rather than projecting a stripe, or pattern, requires extra mechanical complexity, but can provide a more favourable signal-to-noise ratio. Also, the correspondence problem associated with projected patterns is obviated and, furthermore, some types of detector, such as the analogue lateral-effect photodiode, necessitate a collimated projection. Given that the range sensor is scanning over an angle which defines the field of view, and we wish to reorient that field of view in order to fixate on a useful range feature, the problem then is to eliminate the distortions due to these rapid active sensor movements and the motion of the vehicle, and track those features relative to the vehicle.

The mechatronic design of the sensor, its calibration, and its local processing structure have been presented in detail in previous papers [11,12]. To provide context, Appendices A and B briefly review some of this work. However, the central idea presented in this paper is a new feature extraction and tracking algorithm, where the features are linear segments extracted from the scan. The essence of this algorithm is introduced in Section 2.1. The algorithm presented is fast (minimal latency) and tracks features

[*] *E-mail address:* nep@cs.york.ac.uk (N.E. Pears).

PII: S0921-8890(00)00089-0

with an accuracy and reliability which is essential for close range avoidance manoeuvres. Other robotics research using laser range scanners [16,17] has concentrated on robot localisation within a map, which has a different set of sensor performance requirements. Although our sensor system was designed for collision avoidance, the feature extraction and tracking algorithms presented here are also suitable for many mobile robot navigation applications such as map building and localisation algorithms.

A requirement of the algorithm is that it permits (i.e. is robust to) vehicle motion and rotations of the sensor's body (for active sensing). Section 2 discusses the applicability of the extended Kalman filter and argues that the algorithm presented is suitable for any scanning range sensor which provides noisy range measurements at deterministic orientations when the relative motion between sensor and scene is also noisy. Section 3 presents the detailed analysis and modelling that is required for the algorithm. This includes a solution to obtaining a fast and accurate initial feature state estimate. Section 4 discusses feature track management over multiple scans, whilst Section 5 presents results from the extraction/tracking algorithm. Section 6 discusses about extending the feature set to include ellipses.

1.1. Definition of frames of reference

The sensor used in our experiments makes a horizontal one-dimensional scan with a collimated laser about a *projection axis*. This scan defines the size of the sensor's field of view. Also, the whole sensor body can rotate about the sensor axis, which allows reorientation of the sensor field of view in the same plane as the laser scan. Due to the sensor design, the sensor axis and projection axis are not coincident (see Appendix A for more details), and it is necessary to define a projection frame and sensor frame as shown in Fig. 1. Note that the sensor frame is defined in a vehicle frame which in turn is defined in a global frame.

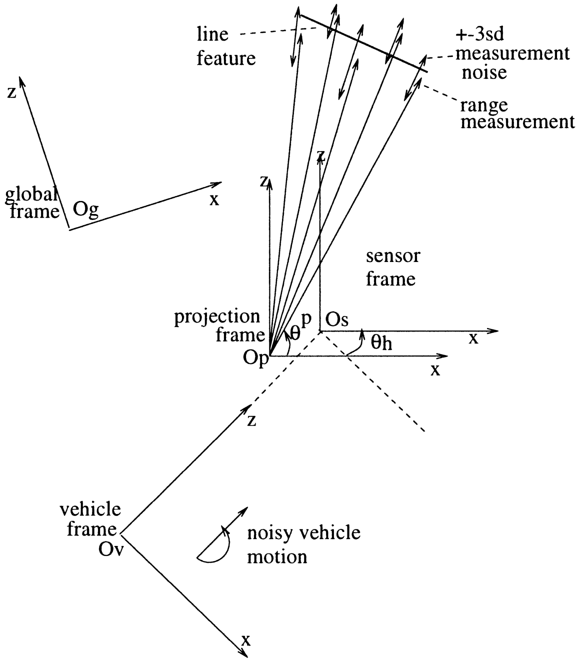


Fig. 1. Frames of reference.

Range features will be extracted and tracked in the vehicle frame, which is sensible for obstacle avoidance, since the vehicle perimeter is expressed in this frame. For our robot, the vehicle frame is located in a global frame by its bar code scanning (BCS) triangulating system. The range features and associated covariances extracted can easily be transferred into that frame for global tasks such as map building. Fig. 1 shows the spatial relationship between the four frames and Table 1 summarises the representations and tasks handled in each of the frames. Note that: (a) Wherever ambiguity could exist, the superscripts p, s, v, g will denote measurements made with

Table 1
Mobile robot reference frames

Frame	Representations	Tasks
Global, O_g	BCS location	Strategic path planning, map building
Vehicle, O_v	Vehicle dimensions, sensor locations	Obstacle avoidance, feature extraction/tracking
Sensor, O_s	Calibration table	Calibration
Projection, O_p	r, θ measurements	Feature observation

respect to one of these four frames. (b) We use the (x, z) plane in each case, the y coordinate would be used if two-dimensional laser scans were employed.

2. Feature extraction and tracking using the EKF: Overview

The Kalman filter is a state estimator for linear systems. When a nonlinear system is effectively modelled by linear equations (linearised), the resulting estimator is called an extended Kalman filter. The fundamental difference between a standard least squares (LS) estimator and the Kalman filter is that the former can only be employed for the estimation of a static state, whereas the Kalman filter can estimate a dynamic state [1]. In designing a Kalman filter estimator, one needs to define (a) a *process model* (the system dynamics), which describes the evolution of the state with respect to some input(s), and subject to some processes noise, and (b) a *measurement model*, which describes how measurements (subject to additive measurement noise) relate to the system state. The success of a particular estimator is crucially dependent on the representations and modelling of both the system dynamics and sensing (measurement). The Kalman filter has been used extensively in Robotics applications for a variety of purposes, such as data fusion [4] and feature tracking [2]. Here, we are particularly concerned with finding good representations for an integrated feature extraction and tracking process for scanning rangefinders.

It is assumed that the range data set for a given scan can be associated with a piecewise linear model of the world. For many real environments in which autonomous vehicles operate (containing, e.g. walls, pillars and boxes) this assumption is not unrealistic. The feature extraction algorithm must estimate the parameters of the line segments, the position of range discontinuities, and their associated uncertainties. In Section 1.1, it was pointed out that features must be extracted and tracked in the vehicle frame. This is because of the following:

- It removes the need for the rest of the system to know the details of the sensor head positioning of the active sensor.
- It is the appropriate frame in which to use the features for implementing obstacle avoidance manoeuvres.
- It is the appropriate frame in which to integrate the (abstracted) data from several different sensor sources in a sensor suite.

We elected to examine recursive (point by point as opposed to batch) approaches to ensure accurate compensation of the vehicle and sensor head motion between range sample instants, and to ensure a feature set that is as up-to-date as possible. This, in addition to adding measurement to feature matching, avoids latency in sensor head control. An approach which is relatively fast would be the standard formulation of a recursive least squares (RLS) estimator. RLS could not be applied directly to (x, z) measurements in the sensor frame, since these two coordinates are not independent, but are related through the laser scan angle (in fact x needs to be deterministic). It is possible to remove this dependence by appropriate transformations of coordinate frames. However, we would still be left with an estimator that estimates a static (non-evolving) state. Clearly, for a moving vehicle, feature extraction and tracking in the vehicle frame requires estimation of a dynamic state. This implies the use of a Kalman filter, and since the robot/sensor system is modelled by a set of articulated rotations, it is formulated in its linearised or extended form (EKF). A polar representation of line segments (length and orientation of line normal) is employed as this seems a natural choice in a scenario, where both the robot and sensor head can rotate. The main benefits the EKF provides are:

1. The EKF's mechanism for validating observations provides a mean for determining range discontinuities (i.e. new features).
2. The process (dynamic state) in the EKF automatically compensates for vehicle movements during feature estimation.
3. The process also allows tracking of all features. (This includes those that have disappeared from the field of view, which gradually degrade, and unless reobserved, are eventually removed from the tracking list.) In our application, feature tracking is used for obstacle avoidance, and repositioning of the sensor head.

In summary, the algorithm described in Section 3 extracts and tracks polar line segments from sequences of noisy range measurements, when the vehicle motion estimate is noisy, and the orientation of the range measurement relative to the robot body is assumed to be noise free (deterministic).

2.1. Tracking features over multiple scans

The EKF process allows compensation for vehicle movement within a single sensor sweep, but also allows the localisation of features to improve across multiple laser sweeps by matching individual range measurements to features. This requires careful management of line segment boundaries in the filter in order to predict feature visibility prior to association of a range measurement and feature. A feature is described as visible at step k if the laser projection direction is predicted to intersect the feature within its boundary points (start point and end point).

Since we wish to compensate for vehicle motion as the sensor sweeps, the EKF must cycle at the same frequency as range measurement are made. This means that the observation in the EKF can only operate on a representation of the infinite, ideal line. Boundary points and their derivatives, length and midpoint, are not directly observed. However, boundary points are fundamental for vehicle and sensor head control, and are also required to predict feature visibility when matching new observations to features extracted from previous scans.

We use the state prediction of the EKF to propagate motion of the boundary points and their associated uncertainty due to the noisy system process (vehicle motion). Describing a boundary point simply in terms of its Cartesian coordinates does not contain any information about the direction in which the line segment was first observed. This means that it is not possible to update the position of the boundary point when observations of the infinite line segment change the estimated polar parameters of that infinite line.

A natural way to solve this problem is to propagate three (infinite) lines, each represented by a polar parameterisation in the vehicle frame. One of these lines is the line feature, and the other two lines delimit the extent of the feature, i.e. their intersections with the line feature define the feature boundaries, which are used in feature visibility computations.

Thus, a novel feature of this algorithm is that the state of a line segment feature is *redundantly* defined as three infinite lines, which is a redundant representation since the bounding rays and line feature itself require a total of six parameters to define rather than the four required to define end points of the line segment feature. Each of the six infinite lines parameters evolve through a predictive, noisy motion process, whereas only the two line segment parameters (not the bounding ray parameters) can be corrected at *every* range measurement through the full EKF machinery (see Fig. 2).

Although each range reading updates an ideal line, it may or may not update one of its associated bounding rays. If the feature is being resized (by growth or erosion as described in Section 4), then a bounding ray is set in the vehicle frame, and assigned zero covariance since angular measurements of the laser projection are considered to be deterministic in the vehicle frame. If range measurements are made (and validated) within the predicted bounding rays of the feature, then no observation of a bounding ray is made. Note that the uncertainty in an unobserved bounding ray increases at each time step due to noise in the vehicles motion (process noise).

In [17], the predictive–corrective structure of the Kalman filter was employed for mobile robot localisation using a range sensor, although here it is described as a “statistical evolution uncertainty technique”. In this work, planar surfaces are extracted from range data using a method based on the Hough transform, and subsequently they are matched to predicted surfaces from a map. In the work presented here, there are a number of differences. Since segmentation and matching measurements to features are closely bound within the same algorithm, which is operating at range sampling rate (rather than at frame capture rate), measurement to feature matching is taking place at the earliest possible time. This gives two benefits: Firstly, it allows motion during the finite capture time between individual range measurements to be accurately compensated for. Secondly, it allows one to match every range measurement to the best possible estimate of the tracked feature. Effectively matching takes place at a lower level, closer to the actual range measurements, and as soon as they are made. This approach can only be effective if an accurate estimate of measurement noise is available at each range sample. Often, as in our case, it can be determined from the intensity image, as described in Appendix B and [11].

In the following section, we describe the process and measurement models for extracting and tracking lines. We describe these with reference to the ideal, infinite line, as this is the part of the state that uses

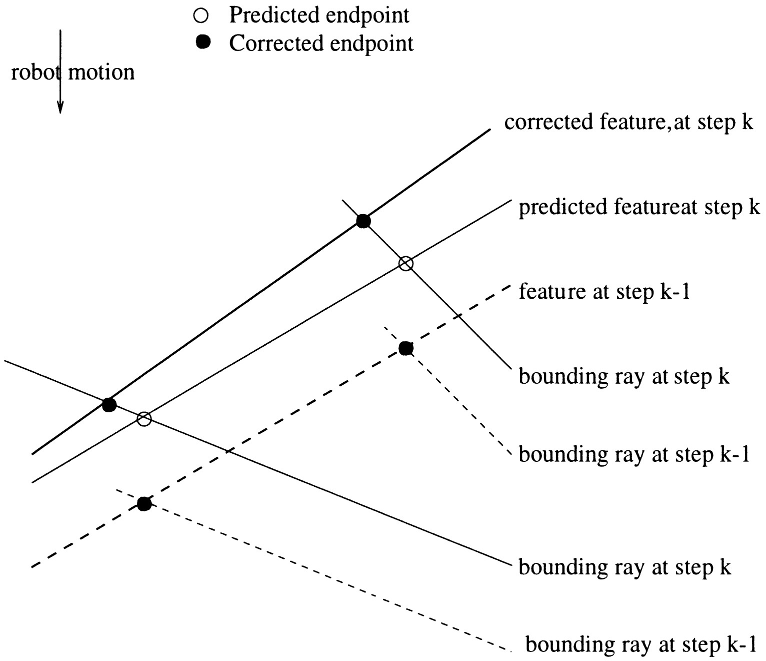


Fig. 2. Line and end point update.

the full EKF machinery. The state prediction is also applied to the bounding rays of the feature.

Assuming that we have an initial state estimate of a line feature and its associated covariance matrix (which is obtained as described in Section 3.6), Sections 3.1–3.5 describe the computations at each stage in the EKF cycle each time a new range measurement is made, namely: state prediction, covariance of the state prediction, observation prediction, observation validation, and finally, state and covariance update.

3. Model analysis for EKF implementation

3.1. State prediction

In the (extended) Kalman filter, a process model describes how the state vector, $\boldsymbol{x}$, evolves with an input vector, $\boldsymbol{u}$, subject to an additive noise vector, $\boldsymbol{v}$. This model has the general form [1]:

$$\begin{aligned} &\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{v}(k), \\ &\boldsymbol{v}(k) \sim N(\mathbf{0}, \boldsymbol{Q}(k)), \end{aligned} \tag{1}$$

where $\boldsymbol{f}$ is the state transition function, and the notation $\boldsymbol{v}(k) \sim N(\mathbf{0}, \boldsymbol{Q}(k))$ denotes that the noise source is modelled as zero-mean Gaussian with covariance $\boldsymbol{Q}(k)$.

We wish to extract and track a variable length list of line features $\{i = 1, \dots, n\}$ whose states are described by polar parameterisations in the vehicle frame, $\boldsymbol{x}_i = [\rho^{\mathrm{v}}, \phi^{\mathrm{v}}]_i^{\mathrm{T}}$, where ρ^{v} is the length of the line normal to the origin and ϕ^{v} is the orientation of the line normal, and the superscript, v, denotes the vehicle frame. The change in $\boldsymbol{x}_i$ over time is due to motion of the vehicle, which over a scan interval (= 0.1 s) is modelled by a constant linear and angular velocity, $(V_R, \dot{\theta}_R)$. If we denote the range sampling period as Δt (= 0.4 ms), which is the time between instances k and $k+1$ in Fig. 3, then the vehicle motion is

$$\Delta\theta^{\mathrm{v}}(k) = \dot{\theta}_R \Delta t, \tag{2}$$

$$\Delta x^{\mathrm{v}}(k) = \frac{-V_R(1 - \cos\Delta\theta)}{\dot{\theta}_R}, \tag{3}$$

$$\Delta z^{\mathrm{v}}(k) = \frac{V_R \sin\Delta\theta}{\dot{\theta}_R}. \tag{4}$$

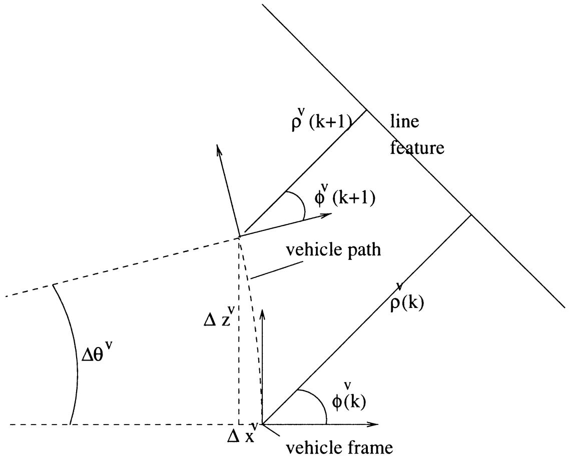


Fig. 3. The EKF process.

Note that if $\dot{\theta}_R$ is small compared to V_R (i.e. if the radius of turning curvature is sufficiently large), then $\Delta x^{\mathrm{v}}(k) \approx 0$ and $\Delta z^{\mathrm{v}}(k) \approx V_R \Delta t$. Fig. 3 also shows that the change in state variable ρ^{v} is given by resolving the vehicle motion $(\Delta x^{\mathrm{v}}(k), \Delta z^{\mathrm{v}}(k))$ in the direction of the normal direction ϕ^{v}, pointing towards the line feature:

$$\rho^{\mathrm{v}}(k) - \rho^{\mathrm{v}}(k+1) = \begin{bmatrix} \Delta x^{\mathrm{v}}(k) \\ \Delta z^{\mathrm{v}}(k) \end{bmatrix} \cdot \begin{bmatrix} \cos\phi^{\mathrm{v}}(k) \\ \sin\phi^{\mathrm{v}}(k) \end{bmatrix}. \quad (5)$$

Hence the state prediction equations at step $k+1$ using measurements up to time k is given as

$$\begin{bmatrix} \hat{\rho}^{\mathrm{v}}(k+1|k) \\ \hat{\phi}^{\mathrm{v}}(k+1|k) \end{bmatrix} = \begin{bmatrix} \hat{\rho}^{\mathrm{v}}(k|k) - \Delta x^{\mathrm{v}}(k)\cos\hat{\phi}^{\mathrm{v}}(k|k) - \Delta z^{\mathrm{v}}(k)\sin\hat{\phi}^{\mathrm{v}}(k|k) \\ \hat{\phi}^{\mathrm{v}}(k|k) - \Delta\theta^{\mathrm{v}} \end{bmatrix}. \quad (6)$$

Note that in this section, we distinguish the true value of a variable and an estimated value, with the "hat" symbol. Thus ϕ may define a line orientation in a model definition, whereas $\hat{\phi}$ defines the estimate of a line orientation within the EKF algorithm.

3.2. Covariance of the state prediction

The covariance of the prediction is then computed as

$$\boldsymbol{P}(k+1|k) = \nabla\boldsymbol{f}\boldsymbol{P}(k|k)\nabla\boldsymbol{f}^{\mathrm{T}} + \boldsymbol{Q}^{\mathrm{v}}(k), \quad (7)$$

where $\nabla\boldsymbol{f}$ is the Jacobian of the state transition function given in Eq. (6), namely

$$\nabla\boldsymbol{f}(k) = \begin{bmatrix} 1 & \Delta x^{\mathrm{v}}(k)\sin\hat{\phi}^{\mathrm{v}}(k|k) - \Delta z^{\mathrm{v}}(k)\cos\hat{\phi}^{\mathrm{v}}(k|k) \\ 0 & 1 \end{bmatrix}. \quad (8)$$

The noise on the process derives from noise in the state estimate of the vehicle motion. In general, this estimate may be derived externally from another sensor system, such as a scanning bar code system, or (more typically) internally by measuring wheel speeds and, if appropriate, steering angles. Although wheel speeds and steering angles can be measured accurately relative to the robot body, the motion that this gives in the environment is noisy due to an irregular floor surface and slip and compliance in the vehicle tyres. This noise is difficult to model analytically and appropriate noise values are best found by empirically tuning the filter. We assume that the noise on the vehicle motion estimate is uncorrelated Gaussian, empirically estimated as

$$\boldsymbol{Q}^R = \begin{bmatrix} \sigma^2_{V_R} & 0 \\ 0 & \sigma^2_{\dot{\theta}_R} \end{bmatrix}. \quad (9)$$

This can be transformed to noise on the state prediction as

$$\boldsymbol{Q}^{\mathrm{v}} = \nabla\boldsymbol{t}\boldsymbol{Q}^R\nabla\boldsymbol{t}^{\mathrm{T}}, \quad (10)$$

where $\nabla\boldsymbol{t}$ is the Jacobian of the transform from the vehicle motion state to the line feature state. Using the approximation $\Delta x^{\mathrm{v}} \approx 0$ and Eq. (6) we have

$$\nabla\boldsymbol{t} = \begin{bmatrix} -\Delta t\sin\hat{\phi}^{\mathrm{v}} & 0 \\ 0 & -\Delta t \end{bmatrix}. \quad (11)$$

Hence

$$\boldsymbol{Q}^{\mathrm{v}} = \begin{bmatrix} (\Delta t\sigma_{V_R}\sin\hat{\phi}^{\mathrm{v}})^2 & 0 \\ 0 & (\Delta t\sigma_{\dot{\theta}_R})^2 \end{bmatrix}. \quad (12)$$

3.3. Observation prediction

The measurement model has the form

$$r(k) = h(\boldsymbol{x}(k), \theta^{\mathrm{p}}, \theta_{\mathrm{h}}) + w(k), \\ w(k) \sim N(0, R(k)), \quad (13)$$

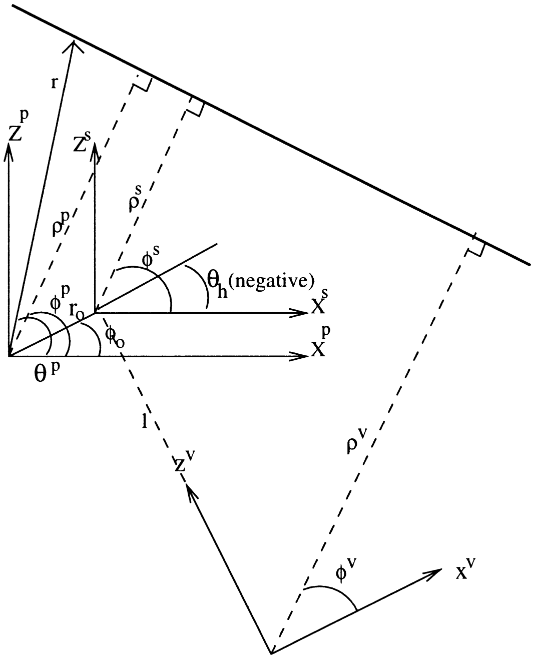


Fig. 4. Articulated polar measurement of a polar line.

where the measurement function, h, relates the state, the inputs (laser projection angle, θ^{p}, and sensor head angle, θ_{h}) and the observation, r (range). Note that w is additive, zero-mean, Gaussian observation noise with variance, R. Thus given a one step ahead predicted state, we can predict an observation as

$$\hat{r}(k+1|k) = h(\hat{\boldsymbol{x}}^{\mathrm{v}}(k+1|k), \theta^{\mathrm{p}}, \theta_{\mathrm{h}}). \tag{14}$$

In addition to a polar state, we consider a polar formulation of the measurement equation since the scan angle, θ^{p}, and sensor head angle, θ_{h}, can be considered to be deterministic in the vehicle frame. Effectively, the measurement model is the articulated polar measurement of a polar line as shown in Fig. 4. To predict the observation, we have to transform the current state estimate from the vehicle frame to the projection frame via the sensor frame. Examination of Fig. 4 reveals the four relations

$$\phi^{\mathrm{p}} = \phi^{\mathrm{s}} = \phi^{\mathrm{v}} - \theta_{\mathrm{h}}, \tag{15}$$

$$\rho^{\mathrm{s}} = \rho^{\mathrm{v}} - l \sin\phi^{\mathrm{v}}, \tag{16}$$

$$\rho^{\mathrm{p}} = \rho^{\mathrm{s}} + r_0 \cos(\phi^{\mathrm{p}} - \phi_0), \tag{17}$$

$$r = \frac{\rho^{\mathrm{p}}}{\cos(\phi^{\mathrm{p}} - \theta^{\mathrm{p}})}, \tag{18}$$

where $(0, l)$ defines the position of the sensor in the vehicle frame and (r_0, ϕ_0) defines the polar position of the sensor frame in the projection frame. If we redefine the input angles as

$$\alpha = \theta_{\mathrm{h}} + \theta^{\mathrm{p}}, \qquad \beta = \theta_{\mathrm{h}} + \phi_0, \tag{19}$$

then combining Eqs. (15)–(19), we can form an observation prediction as

$$\hat{r}(k+1|k) = \frac{\hat{\rho}^{\mathrm{v}}(k+1|k) - l\sin\hat{\phi}^{\mathrm{v}}(k+1|k) + r_0\cos(\hat{\phi}^{\mathrm{v}}(k+1|k) - \beta)}{\cos(\hat{\phi}^{\mathrm{v}}(k+1|k) - \alpha)}. \tag{20}$$

3.4. Observation validation

A matching or data association procedure is required to establish whether the current observation lies on a particular visible feature. Such observation validation is implemented using a validation gate, which normalises the square of the difference between the actual and the predicted observation (the innovation) with the covariance of this quantity, and then tests whether it is below a threshold, g^2. Such a test indicates that the actual observation lies within g standard deviations of the predicted observation and if successful, the observation can be used to update the feature state estimate. The innovation is defined as the difference between the actual and the predicted observation:

$$\nu(k+1) = r(k+1) - \hat{r}(k+1|k). \tag{21}$$

The innovation covariance is obtained by linearising Eq. (13) about the prediction. We then square and take expectations to give

$$\begin{aligned} S(k+1) &= E\left[\nu^2(k+1)\right] \\ &= \nabla\boldsymbol{h}(k+1)\boldsymbol{P}(k+1|k)\nabla\boldsymbol{h}^{\mathrm{T}}(k+1) + R(k+1). \end{aligned} \tag{22}$$

Here

$$\nabla\boldsymbol{h} = \left[\frac{\partial r}{\partial\rho^{\mathrm{v}}}, \frac{\partial r}{\partial\phi^{\mathrm{v}}}\right], \tag{23}$$

where, in our application

$$\frac{\partial r}{\partial \rho^{\mathrm{v}}} = \frac{1}{\cos(\phi^{\mathrm{v}} - \alpha)}, \tag{24}$$

$$\frac{\partial r}{\partial \phi^{\mathrm{v}}} = \frac{-l\cos\phi^{\mathrm{v}} - r_0 \sin(\phi^{\mathrm{v}} - \beta)}{\cos(\phi^{\mathrm{v}} - \alpha)} + \frac{(\rho^{\mathrm{v}} - l\sin\phi^{\mathrm{v}} + r_0\cos(\phi_{\mathrm{v}} - \beta))\sin(\phi^{\mathrm{v}} - \alpha)}{\cos^2(\phi^{\mathrm{v}} - \alpha)}. \tag{25}$$

To generate $R(k+1) = \sigma_r^2(k+1)$, the zero-mean Gaussian noise computed for depth, z, using measured image position noise (Appendix B) and the calibration table for image position to depth [11], must be transformed to a noise on the measured range. In the projection frame, the scan angle is considered to be deterministic, thus

$$\sigma_r^2 = \sigma_z^2 \operatorname{cosec}^2 \theta^{\mathrm{p}}. \tag{26}$$

Having computed the covariance of the innovation, association of the observation with the feature state is achieved through a validation gate test, which is defined by

$$\frac{\nu^2(k+1)}{S} \leqslant g^2. \tag{27}$$

3.5. State and covariance update

If an observation is validated by Eq. (27), the measurement and its variance is used to update the feature state and associated covariance matrix. The well-known information form of the Kalman filter [1] is used giving the state update for validated observations as

$$\hat{\boldsymbol{x}}(k+1|k+1) = \hat{\boldsymbol{x}}(k+1|k) + \boldsymbol{W}(k+1)\nu(k+1) \tag{28}$$

and covariance update as

$$\boldsymbol{P}(k+1|k+1) = \boldsymbol{P}(k+1|k) - \boldsymbol{W}(k+1)S(k+1)\boldsymbol{W}^{\mathrm{T}}(k+1), \tag{29}$$

where W is the Kalman gain, which is computed as

$$\boldsymbol{W}(k+1) = \boldsymbol{P}(k+1|k)\nabla\boldsymbol{h}^{\mathrm{T}} S^{-1}(k+1). \tag{30}$$

3.6. The bootstrap process

When a consecutive sequence of n (typically five) observations are unmatched to an existing feature track, a new feature is instantiated. An n point batch initialisation process is required to provide the initial feature state estimate, $\hat{\boldsymbol{x}}(k|k)$, and associated covariance, $\boldsymbol{P}(k|k)$, on which the recursive process can operate. Often, in Kalman filter implementations, a very rough estimate of the initial state is made and the initial covariance matrix is made large, to reflect this. Here we can compute an initial state, and its associated covariance as we know the variance associated with each range measurement. In one approach, the problem may be formulated as a nonlinear least squares minimisation with $[\rho^{\mathrm{v}}, \phi^{\mathrm{v}}]^{\mathrm{T}}$ chosen to solve

$$[\hat{\rho}^{\mathrm{v}}, \hat{\phi}^{\mathrm{v}}]^{\mathrm{T}} = \underset{\rho^{\mathrm{v}},\phi^{\mathrm{v}}}{\operatorname{argmin}} \sum \left[r_i - \frac{\rho^{\mathrm{v}} - l\sin\phi^{\mathrm{v}} + r_0\cos(\phi^{\mathrm{v}} - \beta_i)}{\cos(\phi^{\mathrm{v}} - \alpha_i)} \right]^2. \tag{31}$$

However, it is not desirable to implement an iterative solution to this equation in an algorithm which must run in real time. A preferable approach is transform the problem into a more tractable linear form. As with the recursive process, it is inappropriate to use a batch LS estimator directly to estimate the initial state, since the world coordinates x and z are not independent, but are related by the scan angle. This means that errors in the measured range will, in general, project into both the x and z coordinates. To minimise the projection of errors into the x coordinate, which allows a standard batch LS to be applied, the following algorithm is implemented.

3.6.1. Initialisation algorithm

1. Transform measurements $[x, z]_i^{\mathrm{p}}$, $i = 1, \ldots, n-1$, to projection frame n. This compensates for both vehicle movements and sensor head movements during the initialisation process. Over the initialisation period (typically 2 ms) the small vehicle movement is treated as deterministic.
2. Compute the centroid, $[\bar{x}, \bar{z}]^{\mathrm{p}}$, of the data set in projection frame n and determine the angle, θ_{c}, of the centroid with respect to the z^{p} axis of this frame.

3. Apply a rotation matrix to the n initialisation points so that the centroid is coincident with the z^{p} axis.
4. Compute a standard weighted least squares on the transformed points to give a standard Cartesian ($z = ax + b$) representation of the line segment.
5. Transform the above representation to a polar representation, $[\hat{\rho}^{\mathrm{p}}, \hat{\phi}^{\mathrm{p}}]^{\mathrm{T}}$, in the projection frame such that

$$\hat{\phi}^{\mathrm{p}} = \tan^{-1}\hat{b} - \theta_{\mathrm{c}} + \tfrac{1}{2}\pi, \qquad \hat{\rho}^{\mathrm{p}} = z_m^{\mathrm{p}}\cos(\tan^{-1}\hat{a}). \tag{32}$$

6. Compute the covariance associated with this estimate as

$$\boldsymbol{P}(k|k) = \left[\nabla\boldsymbol{h}^{\mathrm{T}}\boldsymbol{R}^{-1}\nabla\boldsymbol{h}\right]^{-1}. \tag{33}$$

In the above, $\nabla\boldsymbol{h}$ is the stacked Jacobian measurement matrix

$$\nabla\boldsymbol{h} = \begin{bmatrix} \dfrac{\partial r(\theta_1^{\mathrm{p}})}{\partial\rho^{\mathrm{p}}} & \dfrac{\partial r(\theta_1^{\mathrm{p}})}{\partial\phi^{\mathrm{p}}} \\ \vdots & \vdots \\ \dfrac{\partial r(\theta_n^{\mathrm{p}})}{\partial\rho^{\mathrm{p}}} & \dfrac{\partial r(\theta_n^{\mathrm{p}})}{\partial\phi^{\mathrm{p}}} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\cos(\hat{\phi}^{\mathrm{p}} - \theta_1^{\mathrm{p}})} & \dfrac{\hat{\rho}^{\mathrm{p}}\tan(\hat{\phi}^{\mathrm{p}} - \theta_1^{\mathrm{p}})}{\cos(\hat{\phi}^{\mathrm{p}} - \theta_0^{\mathrm{p}})} \\ \vdots & \vdots \\ \dfrac{1}{\cos(\hat{\phi}^{\mathrm{p}} - \theta_n^{\mathrm{p}})} & \dfrac{\hat{\rho}^{\mathrm{p}}\tan(\hat{\phi}^{\mathrm{p}} - \theta_n^{\mathrm{p}})}{\cos(\hat{\phi}^{\mathrm{p}} - \theta_n^{\mathrm{p}})} \end{bmatrix} \tag{34}$$

and $\boldsymbol{R}$ is a diagonal matrix of observation variances

$$\boldsymbol{R} = \begin{bmatrix} \sigma_{r_1}^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_{r_n}^2 \end{bmatrix}. \tag{35}$$

7. Transform the projection frame state into the vehicle frame state using

$$\hat{\rho}^{\mathrm{v}} = \hat{\rho}^{\mathrm{p}} - r_0\cos(\hat{\phi}^{\mathrm{p}} - \phi_0) + l\sin(\hat{\phi}^{\mathrm{p}} + \theta_{\mathrm{h}}), \tag{36}$$

$$\hat{\phi}^{\mathrm{v}} = \hat{\phi}^{\mathrm{p}} + \theta_{\mathrm{h}}. \tag{37}$$

8. Finally, transform the projection frame covariance to the vehicle frame covariance:

$$\boldsymbol{P}^{\mathrm{v}} = \nabla\boldsymbol{u}\boldsymbol{P}^{\mathrm{p}}\nabla\boldsymbol{u}^{\mathrm{T}}, \tag{38}$$

where $\nabla\boldsymbol{u}$ is the Jacobian of the transform between the projection frame and the vehicle frame

$$\nabla\boldsymbol{u} = \begin{bmatrix} 1 & r_0\sin(\hat{\phi}^{\mathrm{p}} - \phi_0) - l\cos(\hat{\phi}^{\mathrm{p}} + \theta_{\mathrm{h}}) \\ 0 & 1 \end{bmatrix}. \tag{39}$$

4. Feature track management

Tracking features relies on computation of the predicted visibility of features, as indicated in Fig. 5. Tracked boundary rays, $[\rho_{\mathrm{b}}^{\mathrm{v}}, \phi_{\mathrm{b}}^{\mathrm{v}}]^{\mathrm{T}}$ and feature lines $[\rho^{\mathrm{v}}, \phi^{\mathrm{v}}]^{\mathrm{T}}$ are intersected to give boundary point coordinates $(x^{\mathrm{v}}, z^{\mathrm{v}})$ as

$$\begin{bmatrix} x^{\mathrm{v}} \\ z^{\mathrm{v}} \end{bmatrix} = \frac{1}{\sin(\phi_{\mathrm{b}}^{\mathrm{v}} - \phi^{\mathrm{v}})}\begin{bmatrix} \sin\phi_{\mathrm{b}}^{\mathrm{v}} & -\sin\phi^{\mathrm{v}} \\ \cos\phi_{\mathrm{b}}^{\mathrm{v}} & \cos\phi^{\mathrm{v}} \end{bmatrix}\begin{bmatrix} \rho^{\mathrm{v}} \\ \rho_{\mathrm{b}}^{\mathrm{v}} \end{bmatrix} \tag{40}$$

which is then transformed into the projection frame using the sensor head angle as

$$\begin{bmatrix} x^{\mathrm{p}} \\ z^{\mathrm{p}} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_{\mathrm{h}} & \sin\theta_{\mathrm{h}} & r_0\cos\phi_0 - l\sin\theta_{\mathrm{h}} \\ -\sin\theta_{\mathrm{h}} & \cos\theta_{\mathrm{h}} & r_0\sin\phi_0 - l\cos\theta_{\mathrm{h}} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x^{\mathrm{v}} \\ z^{\mathrm{v}} \\ 1 \end{bmatrix} \tag{41}$$

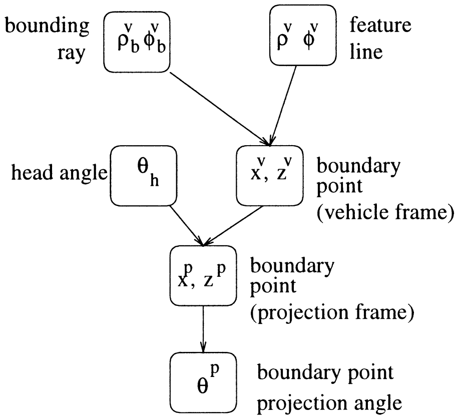


Fig. 5. Visibility prediction.

and finally transformed into a visibility angle in the projection frame, as illustrated in Fig. 5.

For each laser projection angle, an attempt to match the range measurement to visible features along that ray is made. In the simplest case, an observation is made within the predicted visibility angle of the feature and is matched (validated) to that feature, then the parameters and associated covariances of the infinite feature line are updated in the EKF framework. Multiple matches may be found as occlusion is not explicitly modelled. In such cases the match with the smallest innovation is chosen.

Beyond this simplest case, feature track management is required. (We refer to a feature track as the evolution of the redundant six-dimensional feature state over multiple scans.) This entails the addition and removal of features from the tracking list, and manages the features boundaries (start points and end points), as defined by the tracked bounding rays. Feature track management employs six specific operations, described below, which can be conceptually paired as three operation/inverse pairs: add–delete, grow–erode, and split–merge.

- *Feature add*. When a consecutive sequence of n (typically five) observations are unmatched to an existing feature track, a new feature is instantiated. This occurs when there are either no visible features, or, for all visible features, the validation gate test for data to track matching fails.
- *Feature delete*. When either of the infinite line feature variances $(\sigma_\rho^2, \sigma_\phi^2)$ exceeds a threshold, the feature is deleted. (Other heuristic refinements may be made, such as deleting features below a minimum length, above a maximum product of $\sigma_\rho \sigma_\phi$, or above a maximum range from the vehicle.)
- *Feature grow*. When observations are matched beyond, but consecutive to the feature end point, the feature grows from that end point. This requires that the system attempts a feature match to features which have become invisible over the last increment of projection angle.
- *Feature erode*. When observations are not matched to a visible feature immediately after, and consecutive to, the feature start point, the feature is eroded at the start point. Also, when observations fail to match within the scan angle $n\,\delta\theta^{\mathrm{p}}$ from the end point (where $\delta\theta^{\mathrm{p}}$ is the laser projection angle step), the feature is eroded at the end point.
- *Feature split*. If, after a sequence of matched observations, a match fails before the scan angle is within $n\,\delta\theta^{\mathrm{p}}$ of the feature end point, the feature splits into two “child” features, which have the same ideal line parameters, but different boundary points.
- *Feature merge*. When two features are visible at a particular scan angle (or the previous scan angle if a feature is growing), and the observation matches both feature tracks, then a check is made to see if the two features can be merged. This is possible if their parameters (of the ideal line) and associated variances overlap in both ρ and ϕ space. If sufficient overlap is found, they are merged by weighting the two sets of infinite line parameters according to their variances, and selecting the appropriate boundary points. Typically, this occurs when a feature needs to grow, but in the opposite direction to the scanning motion.

5. Performance results

The performance of the system is dependent on the level of noise entering the system. It seems that vehicle orientation noise has a bigger detrimental effect than translation noise. If the estimation of vehicle motion is noisy (e.g. if odometry only is used), then it is likely that less matches between range measurements and features will be made. However, the level of motion noise must be included in the EKF. If this is high, then unmatched feature tracks are quickly deleted, and the robots estimate of its immediate environment is essentially what it has seen over the last few scans.

To test the performance of our algorithms, we used the detailed sensor simulation presented in [13], which gives noisy range scans which are virtually indistinguishable from our sensor. This allows us to use controlled scenes with known ground truth. In our first experiment, the line segments extracted from a scene, with low reflectivity targets parallel to the sensor, are superimposed on the measured points, and plotted in Fig. 6. Here the sensor frame origin and vehicle frame origin are coincident and the results are plotted in a global frame.

The growth of these line segments is plotted in a sequence of frames shown in Figs. 7–14 as the sensor makes its initial scan. In this sequence of figures, we see three new features instantiated, each of which

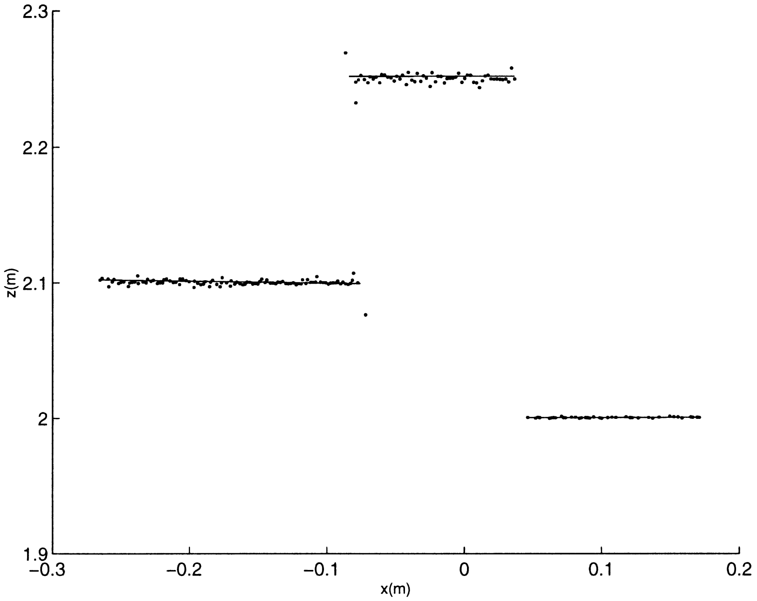


Fig. 6. Measurements and extracted features: sensor at x=0 m, z=1.5 m.

grows as new range measurements are matched to the feature beyond its end point. Notice that, to varying degrees, the line segments extracted are not quite horizontal due to range measurement noise and the small angle subtended by the feature. As features grow, they gradually move towards a horizontal orientation. This is shown more explicitly by examining the plot of orientation in the EKF state (ϕ^{v}) which is shown in Fig. 15. Here we see that, for each new feature, the state estimate made by the feature initialisation algorithm is slightly off the true value, $\frac{1}{2}\pi$ (a horizontal feature in the controlled scene). As more range readings are measured, the state estimate converges on the true value. For completeness, a plot of the remaining variable in the EKF state is given in Fig. 16. (This is the normal displacement of the feature, ρ^{v}.)

The targets used in the above experiment gave step discontinuities in the scene. In other scenarios, there are discontinuities in the *gradient* of the scene. This means that the end point of one feature should be close to the start point of another. At such *ridge* points, matching measurements to the correct feature is significantly more difficult. Figs. 17 and 18 show range scans when the sensor is facing a pair of scene segments, fixating on the ridge discontinuity, and moving towards a wall. In Fig. 17, the sensor origin was at (1.5 m, 0.0 m) in the global frame when the scan was taken. In this case, four segments rather than two are generated, although at a large range in an obstacle avoidance application, this is not critical. The fragmentation may be due to the errors from the linearisation of the sensor geometry becoming large with the large range noise. This fragmentation can be reduced by increasing the validation gate, but this is undesirable as it leads to an increased overshoot at newly observed slope discontinuities (ridges). Overshoot occurs when observations are produced by a new scene segment, but still lie within a growing feature's validation gate. Fig. 18 shows the range measurements and EKF output in the same environment, when the vehicle had moved forwards by 0.5 m. We can see that the algorithm has merged the four edges into two edges, which accurately represent the underlying structure of the environment, and the small overshoot has been reduced.

Extended experimentation with the system has led us to conclude that, in general, the algorithm performs well, but can experience problems of (a) fragmentation at large ranges, and (b) overshoot at newly observed ridge points. Both of these problems are mitigated by feature track management, which will merge features if they become sufficiently close in feature space, and

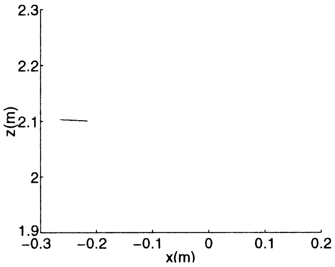


Fig. 7. Frame 1.

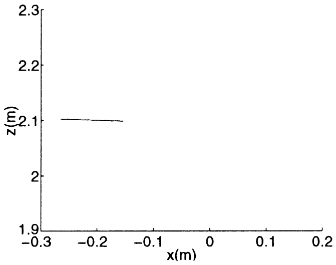


Fig. 8. Frame 2.

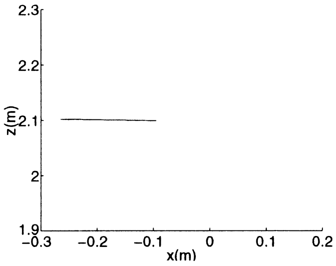


Fig. 9. Frame 3.

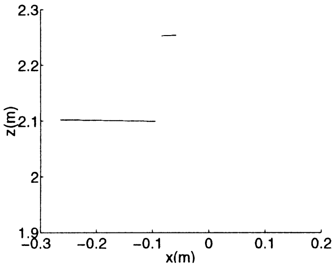


Fig. 10. Frame 4.

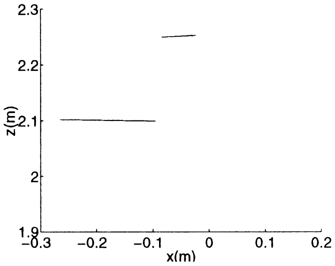


Fig. 11. Frame 5.

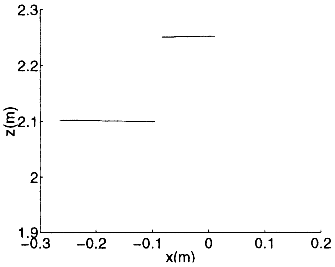


Fig. 12. Frame 6.

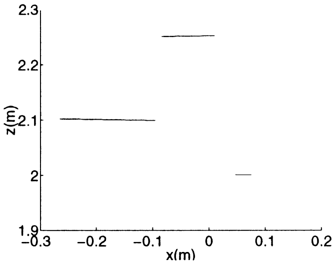


Fig. 13. Frame 7.

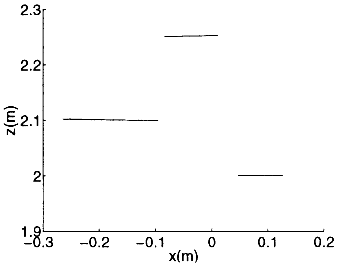


Fig. 14. Frame 8.

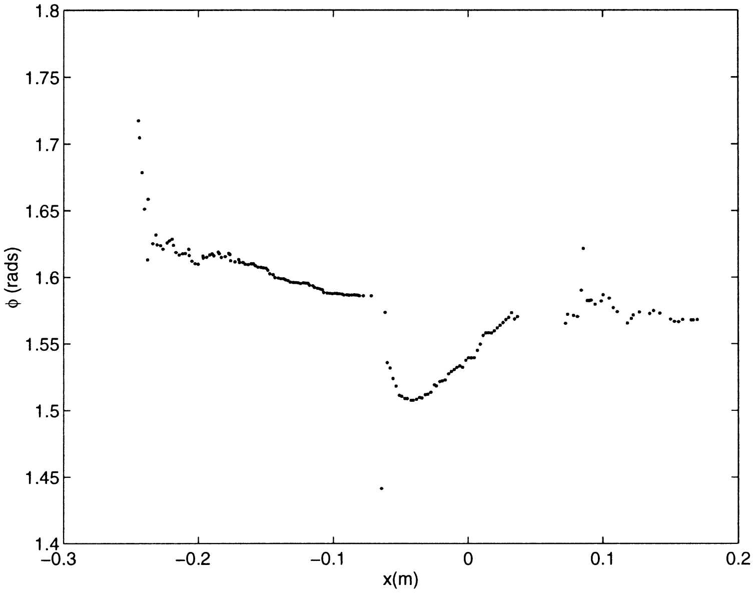


Fig. 15. ϕ^{v} evolution in the EKF: sensor at x=0 m, z=1.5 m.

will reduce overshoot by associating observations with the correct feature track over multiple scans.

We found that overshoot was not a significant problem in our obstacle avoidance application, particularly if ridges were observed over a large number of range scans. However, they may need to be eliminated (as far as possible) in other applications such as map building. One possible approach would be to backtrack to

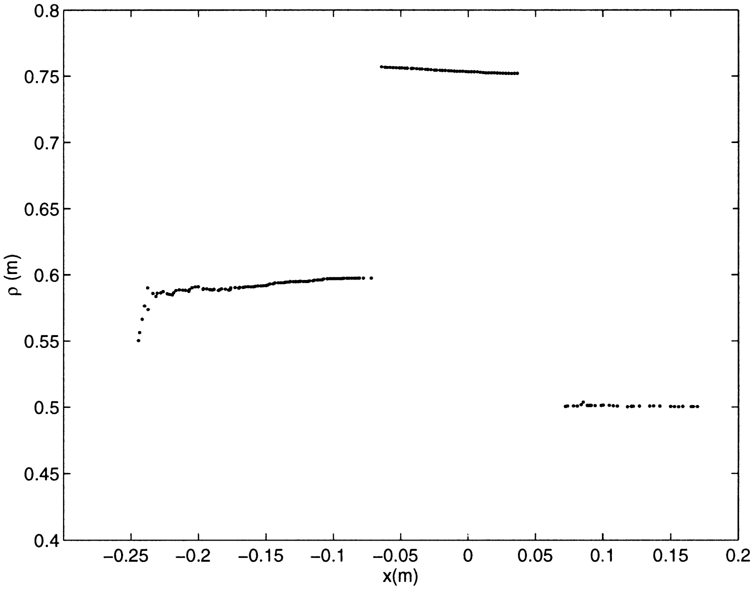


Fig. 16. ρ^{v} evolution in the EKF: sensor at x=0 m, z=1.5 m.

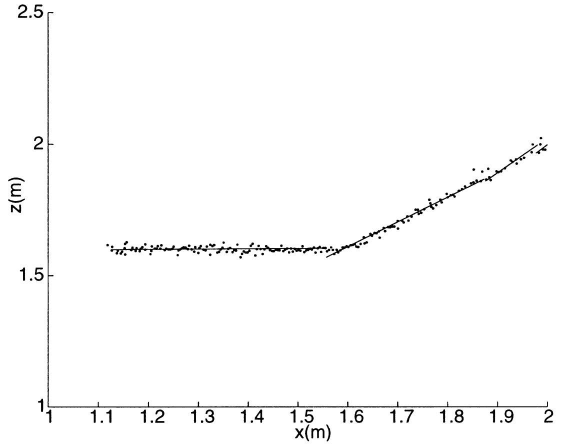


Fig. 17. Measurements and extracted features: sensor at x=1.5 m, z=0.0 m.

a previously saved state to remove the offending outlier(s), correct for vehicle motion in the backtrack period, and, if necessary, reconstruct the ridge points by intersection of extracted line features either side of the ridge.

6. Discussion: Extending the EKF to ellipses

A possible extension to this algorithm is to augment the feature set, so that the environment can be modelled as set of planes and cylinders. The range profile of a cylinder is circular if the cylinder is vertical, and elliptical, otherwise. Since circles are a subset of ellipses, the natural extension from exclusively piecewise-linear features is to allow for elliptical sections. In the batch initialisation process, the appropriate model (linear/elliptical) which best fits the initialisation data could be selected, and the associated EKF measurement equation would be employed for the recursive stage of the algorithm. Unfortunately, the batch stage requires LS fitting of the standard bi-quadratic conic equation [15,19] over small sections of data. This is very ill-conditioned [14]: small changes in the range measurements can change the fit from an ellipse to a parabola or hyperbola, and the observation prediction has a large variance. This leads us to believe that a multi-feature set algorithm would be more suitable to a pre-segmented batch scheme. As a consequence, significant latency could be introduced in the head control. In the EKF algorithm described, elliptical profiles are broken down into a set of line segments. Although this does not accurately reflect the underlying structure of the environment, this situation can be detected as the discontinuities extracted will not exhibit predictable behaviour when the vehicle and sensor head move.

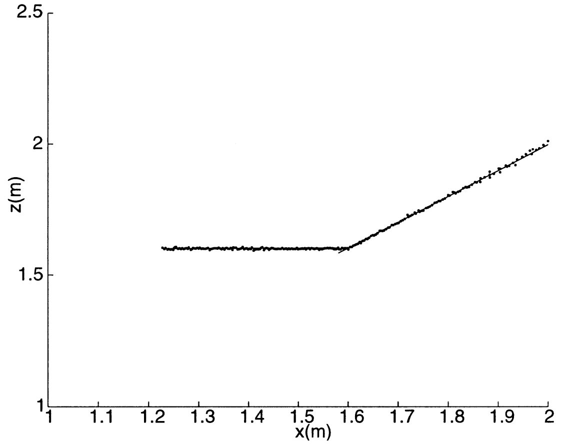
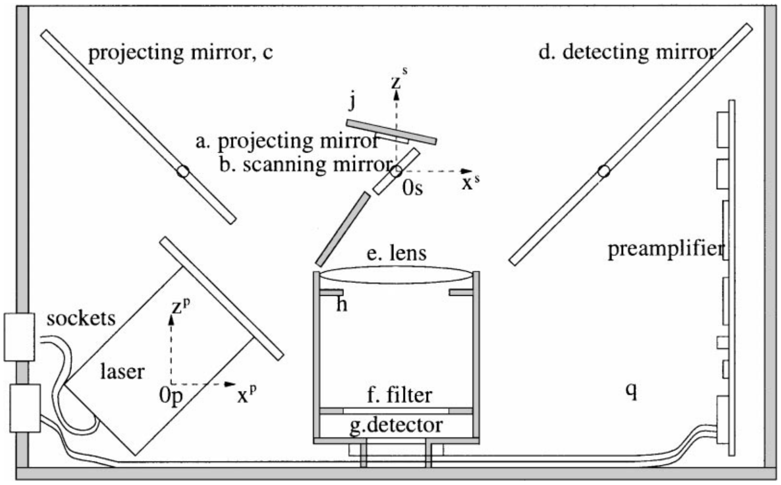


Fig. 18. Measurements and extracted features: sensor at x=1.5 m, z=0.5 m.

7. Conclusions

A line segment extraction and tracking algorithm has been developed based on the EKF. The EKF process models the effect of (noisy) vehicle motion on a polar line representation. The EKF measurement model models the articulated polar observation of the polar line state. Choosing this representation has given trivial line orientation (ϕ) computations but more complex computations for the line normal (ρ). A key idea introduced is the use of a redundant representation of line features, such that a feature is represented by a polar feature line and a pair of polar bounding lines. This preserves information about the direction in which feature end points were observed, which is instrumental in accurate prediction of feature observability. The paper has also described a fast feature track initialisation procedure and has detailed a number feature track management operations.



A.1. Plan view of the sensor head.

Extended experimentation with the system has led us to conclude that, in general, the algorithm performs well, and problems of fragmentation and overshoot are mitigated by feature track management. Also, if necessary, overshoot may be eliminated by heuristic procedures, which "fine tune" the algorithm.

Appendix A. Sensor configuration

This section provides a brief summary of the sensor configuration presented in [11]. The means of scanning the laser and lens in exact synchronism is shown in the plan view in Fig. A.1, which is an adaptation of Livingstone and Rioux's [7] configuration. In this implementation, a collimated laser is scanned in one dimension over twice the angle over which the scanning mirror deflects, and the centre of scanning is at the virtual image point O_p. The lens is effectively scanned, in exact synchronism with the laser, around virtual image point q in the sensor on an arc with radius equal to the separation between the scanning mirror and the lens. The synchronised scanning optics in Fig. A.1 is termed the *sensor head*. This sensor head is mounted on a servo driven platform which can rotate the field of view of the sensor head between $+90^\circ$ and -90° relative to the forward looking direction. Full specifications of the sensor performance are given in Table A.1.

Table A.1.
Sensor specifications

Specification	Value
Repeatability (3σ) at 0.5 m	0.1 cm approximately
Repeatability (3σ) at 1 m	0.4 cm approximately
Repeatability (3σ) at 1.5 m	2.4 cm approximately
Repeatability (3σ) at 2 m	7.6 cm approximately
Field of view	40°
Depth of field	2.1 m
Stand of distance (minimum range)	0.4 m
Maximum range	2.5 m
Range measurements per scan	256
Range measurement frequency	2.5 kHz
Bandwidth of detector	1 kHz
Scan frequency	9.8 Hz
Laser power	0.9 mW
Laser wavelength	670 nm
Laser class	II (maximum 1 mW)
Sensor head position resolution	0.36°
Head response time (90° step)	0.5 s

(Note that repeatability specifications are given for a target of good reflectivity at 670 nm.)

Appendix B. Image position measurement

The geometric means of range measurement described in Appendix A requires the one-dimensional measurement of image position. An analogue means of measurement is provided by a two-terminal device called the lateral-effect photodiode (LEP) which acts as a photocurrent divider so that the position of the light, p, from the centre of a detector of length, P, is

$$p = \frac{I_1 - I_2}{I_1 + I_2}\left(\frac{P}{2}\right), \quad -\tfrac{1}{2}P \leqslant p \leqslant +\tfrac{1}{2}P \tag{B.1}$$

and the detector current, I_0, is the sum of the terminal currents, I_1 and I_2. A previous analysis [11] established a relationship between the standard deviation associated with an image position measurement and the detector current for that measurement as

$$\sigma_{p_n} = \frac{\sigma_p}{\frac{1}{2}P} = \frac{I_n}{I_0}. \tag{B.2}$$

Hence, if we determine I_n from calibration experiments, then every time a measurement is made, image position variance can be computed using the signal strength, I_0. This can be scaled by (the square of) the triangulation gain, which is the magnitude of the local gradient, $|\partial z/\partial p|_{z,\theta}$ in the calibration table to give an estimate of range variance. This variance information is essential to allow robust algorithms to be applied to the raw range data. In particular, it is used in the extended Kalman filter algorithm for simultaneous feature extraction and tracking.

References

[1] Y. Bar-Shalom, X. Li, Estimation and Tracking: Principles, Techniques and Software, Artech House, Boston, MA, 1993.

[2] R. Deriche, O. Faugeras, Tracking line segments, Image and Vision Computing 8 (4) (1990) 261–270.

[3] H.R. Everett, Survey of collision avoidance and ranging sensors for mobile robots, Robotics and Autonomous Systems 5 (1989) 5–67.

[4] J. Horn, J. Neira, J.D. Tardos, G. Schmidt, Fusing range and intensity images for mobile robot localisation, IEEE Transactions on Robotics and Automation 15 (1) (1999) 76–84.

[5] T. Kanade, A. Gruss, L.R. Carley, A very fast VLSI rangefinder, in: Proceedings of the IEEE International Conference on Robotics and Automation, Sacramento, CA, 1991.

[6] R. Kuc, A spatial sampling criterion for sonar obstacle detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7) (1990) 686–690.

[7] F.R. Livingstone, M. Rioux, Development of a large field of view 3D vision system, SPIE Proceedings 665 (1986) 188–194.

[8] H.R. Lo, A. Blake, D. McCowen, D. Konash, Epipolar geometry for trinocular active range sensors, in: Proceedings of the British Machine Vision Conference, 1990.

[9] J.A. Marszalec, H.M. Keranen, A photoelectric range scanner using an array of led chips, in: Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, 1992, pp. 593–598.

[10] G.L. Miller, E.R. Wagner, An optical rangefinder for autonomous robot cart navigation, SPIE Proceedings 852 (1987) 122–134.

[11] N.E. Pears, P.J. Probert, An intelligent active range sensor for mobile robot guidance, Mechatronics 6 (7) (1996) 733–759.

[12] N.E. Pears, An intelligent active range sensor for vehicle guidance, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Osaka, Japan, 1996, pp. 81–88.

[13] N.E. Pears, Modelling of a scanning laser range sensor for robotic applications, Advanced Robotics 13 (5) (1999) 549–562.

[14] J. Porrill, Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter, Image and Vision Computing 8 (1) (1990) 37–41.

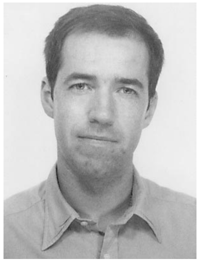
[15] P.L. Rosin, A note on the least squares fitting of ellipses, Pattern Recognition Letters 14 (10) (1993) 799–808.

[16] B. Schiele, J.L. Crowley, F. Wallner, Position estimation using principal components of range data, Robotics and Autonomous Systems 23 (4) (1998) 267–276.

[17] G. Schmidt, J. Horn, Continuous localization of a mobile robot based on 3D-laser-range-data, predicted sensor images, and dead-reckoning, Robotics and Autonomous Systems 14 (2/3) (1995) 99–118.

[18] A.M. Wallace, J. Clark, G.C. Pronzato, Measuring range using a triangulation sensor with variable geometry, IEEE Transactions on Robotics and Automation 14 (1) (1998) 60–68.

[19] W.Y. Wu, Elliptical object detection by using its geometric properties, Pattern Recognition 26 (10) (1993) 1499–1509 .



Nick Pears was awarded a B.Sc. in Engineering Science, and a Ph.D. in Robotics, 1990, by Durham University, UK. He then undertook post-doctoral research in the “Sensors and Architectures” division of the Robotics Research Group, Oxford University. In 1994, he was elected as a College Lecturer and Fellow of Girton College, Cambridge University. In 1998, he joined the Computer Science Department, University of York, UK, in his current Lectureship. His research interests are in Robotics and Computer Vision and include optical sensor design, theoretical modelling of sensors, recursive range feature extraction under motion, real-time architectures for robotics, vehicle guidance using visual cues, human–computer interaction and digital media applications of vision.