Tracking Naturally Occuring Indoor Features in 2D and 3D with Lidar Range/Amplitude Data

M.D. Adams* A. Kerstens†
Institute of Robotics
Swiss Federal Institute of Technology
Zürich, Switzerland

September 21, 1998

Abstract

Sensor Data Processing for the interpretation of a mobile robot's indoor environment, and the manipulation of this data for reliable localisation, are still some of the most important issues in robotics. This article presents algorithms which determine the true position of a mobile robot, based on real 2D and 3D optical range and intensity data. We start with the physics of the particular type of sensor used so that the extraction of reliable, and repeatable information (namely edge coordinates) can be determined taking into account the noise associated with each range sample and the possibility of optical multiple path effects. Again, applying the physical model of the sensor, the estimated positions of the mobile robot and the uncertainty in these positions are determined. We demonstrate real experiments using 2D and 3D scan data taken in indoor environments. In order to update the robot's position reliably, we address the problem of matching the information recorded in a scan to, firstly, an a priori map and secondly, to information recorded in previous scans, eliminating the need for an a priori map.

1 Introduction

An approach to navigation which unifies the extraction of local information, from a vehicle's immediate 3D surroundings, with the application of a simple, mobile robot kinematic, model based localisation algorithm is presented. We begin by analysing 2D and 3D optical range and intensity data from a newly developed *lidar* (*li*ght *d*etection *a*nd *r*anging) sensor. In recent years these sensors have been used more and more in robotics (Johnston A. R. 73; Krotkov E. 90; Miller G. L. and Wagner E. R. 87; Nitzan D. et al. 77) and our experience with purchased sensors has led us to develop our own lidar (Adams 96). By considering the physics involved in this sensing process we briefly show in section 2 the effect of electronic noise and optical multiple path effects upon the sensor data. This is done in order to produce, in section 3, an informed answer to the question "What information should a mobile robot localisation algorithm look for, within the sensor data?" A physical model which statistically relates the noise associated with individual range readings to the expected coordinates of points along indoor line segments is derived. We present and quantify the resulting edge detector using real *lidar* data and show that the edge detection method is an *on line* process. This means that the algorithm does not wait

*M.D. Adams is now employed as a research scientist at ESEC SA, Cham, Switzerland

†A. Kerstens is affiliated with the Control Laboratory, Dept. Electrical Engineering, T.U. Delft, The Netherlands.

for full scans and then determine edges, but instead determines their coordinates immediately after they are sensed.

In section 5, we demonstrate a simple mobile robot kinematic model which relates the true position of the robot to the 2D and 3D detected edge coordinates found in section 3. Under the framework of the extended Kalman filter (EKF) we determine the estimated position and positional uncertainty of the robot immediately after each scan. We show two localisation examples. Firstly the positions of the mobile robot are determined using a simple *a priori* map of the environment. Exactly how this *a priori* map is formed is also explained. In the second example, the positions of the mobile robot are determined *without* any *a priori* map where new edges are compared and matched to those extracted from previous scans. We also comment on the general problem of mobile robot localisation without an *a priori* map.

Finally in section 7, edge coordinates from 3D scans are used to localise a mobile robot. We demonstrate the generality of the algorithms as we match edges determined at different scanning azimuth angles between 3D range scans. Future research issues in the area of 3D robot localisation will also be discussed.

2 Optical Range and Intensity Data

In order to place our work into perspective, we will briefly present here the results of single 2D and 3D *lidar* (light Detection and Ranging) scans recorded in an indoor laboratory environment. The particular sensor used is a lidar, developed by the first author, employing a low powered 1mW, eye safe LED which can measure ranges from almost all types of surface between 0 and 15 m (Adams 96; Borenstein et al. 95). A comparison of this and other commercially available optical range sensors is given in (Adams 92; Borenstein et al. 95; Hebert M. and Krotkov E. 91). A photograph of the 3D scanning sensor used throughout this article is shown in figure 1. It can be seen in the photo that the mirror can be scanned about two axes, namely the horizontal and vertical axes, thus allowing the 1D columnated LED light beam incidence upon the regions of interest within the local environment. The component of the light reflected parallel to the incident beam is then received by the sensor and a range[1]/intensity data pair is produced at the sensor's outputs. Using the scanning mechanism shown in figure 1 then produces either a 2D or 3D scan of the environment as required.

2.1 2D Data Production

Figure 2 shows a 2D range (top) and relative received light intensity (bottom) scan recorded within a laboratory environment. The actual environment is shown as the dotted lines and the range data is shown as the larger dots. In the bottom scan, the relative received signal amplitude as a function of scanning angle can be seen. The larger this signal is the more 'trustworthy' the corresponding range data point is. This will be explained further in section 2.1.2.

2.1.1 Spurious data - mixed pixels

Close examination of figure 2 shows that spurious data points sometimes result immediately before or after detected edges and surface reflectance changes. Between regions A and B in figure 2 for example, the actual range suddenly jumps from 1.62 metres (corresponding to the edge at A) to 3.90 metres at B. The observed range, according to the sensor, changes from 1.62 metres to only 2.67 metres (point C in figure 2). This effect occurs when the transmitted light

[1] The sensor employs the amplitude modulated continuous wave (AMCW) method for range estimation (Borenstein et al. 95)

Figure 1: *A photograph showing the 3D scanning lidar sensor.*

beam is split between two surfaces of differing range and/or reflectivity. Each surface produces its own reflection and the output range reading from the sensor is made up of *both* reflected signals.

In (Adams 92) we have considered in detail the effect of splitting the light beam between two targets. In previous work the cause of the spurious points has not received much attention, and it has simply been stated that they are inherent in any A.M.C.W. optical system and cannot be removed (Hebert M. and Krotkov E. 91). In response to this we have built a detector which has a high success rate at identifying such points when they are caused by *either* reflectance *or* range changes or both simultaneously (Adams and Probert 96). In this article, we use this detector on all range data scans (unless otherwise specified) before further processing the data.

2.1.2 Range Uncertainty - Determining the Range Variance

The bottom scan in figure 2 shows the amplitude of the received signal throughout the full 360^o scan. This information is very useful as it tells us the uncertainty associated with each corresponding range reading. To derive a relationship between the amplitude of the received light signal, V_r and the range uncertainty or more precisely the range variance, σ_r^2, it is necessary to gain an understanding of the physics involved when the light is reflected from a point in the environment and converted into a range voltage by the sensor's receiver. An explanation of the propagation of noise within the sensor and a derivation of the relationship between σ_r^2, the range variance and V_r, the signal amplitude is given in (Adams and Probert 96; Adams 92; Connor

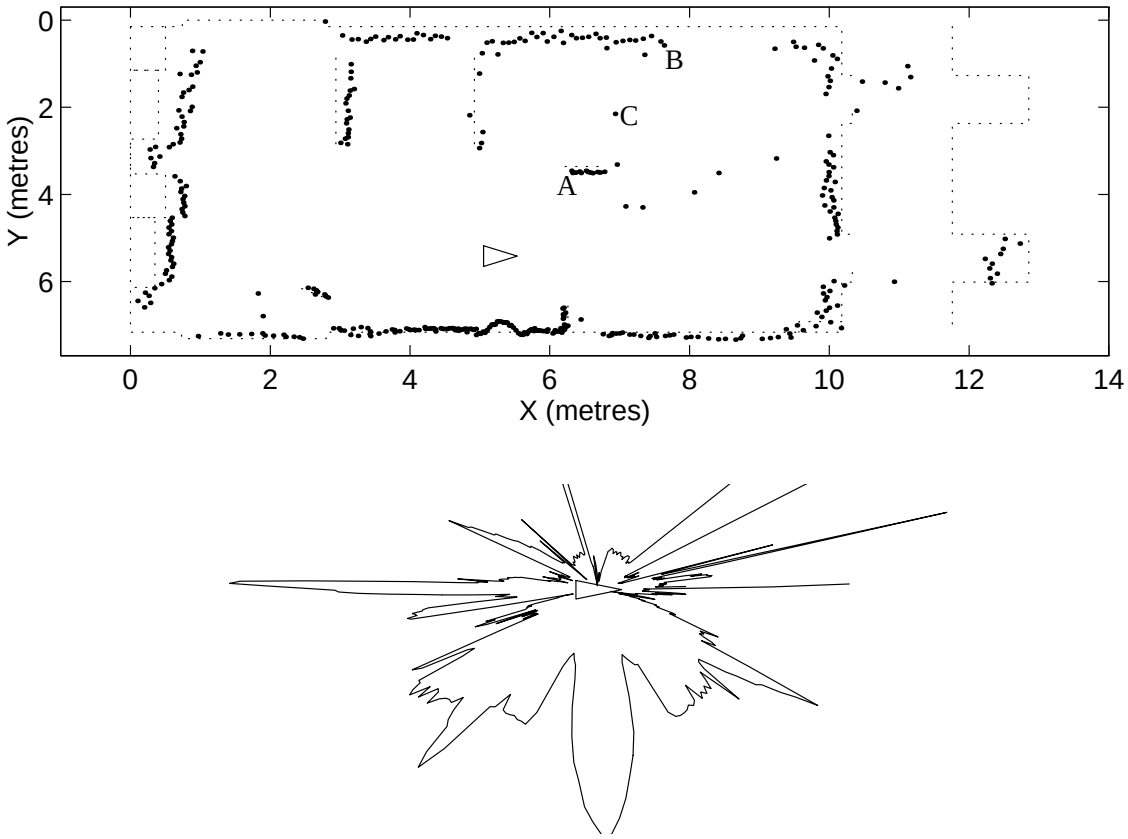

Figure 2: *2D range (top) and received signal amplitude (bottom) scans recorded within a laboratory environment. The triangle represents the sensor's position and the dotted lines in the top scan represent the actual environment. The top scan shows range as a function of scanning angle (larger dots) and the bottom scan shows the relative received signal amplitude as a function of scanning angle.*

F. R. 82). We repeat the relationship here as it will be very useful in sections 3.1 and 5 where we process the range data according to its variance to find edges and localise a mobile robot respectively. From (Adams and Probert 96):

$$\sigma_r^2 \approx \left(\frac{\lambda\sigma_n}{4\pi}\right)^2 \left(\frac{1}{V_r}\right)^2 + \sigma_e^2 \tag{1}$$

where λ is the signal modulation wavelength, σ_n^2 is the constant variance of the initial electronic noise source before phase detection and σ_e^2 is the additive electronic noise effect due to the constant noise variance within the reference signal.

Tests with the sensor gave the numerical relationship between σ_r^2 and V_r as:

$$\sigma_r^2 \approx \frac{729.124}{V_r^2} + 83.000 \tag{2}$$

where V_r is measured in volts and σ_r^2 is the variance $\times 10^{-6}$ m^2. By direct comparison with equation 1 we see that for $\lambda = 60$ metres:

$$\sigma_n^2 \approx 5.655 \times 10^{-3} \text{ (volts)}^2, \quad \sigma_e^2 = 83.000 \times 10^{-6} \text{ (metres)}^2 \tag{3}$$

Therefore, for each observed range estimate we now have an approximate estimate of the range variance.

2.2 3D Data Production

With the scanning system shown in figure 1 it is possible to represent the 3D range data from the sensor in Cartesian form as in figure 3. In this representation one can clearly see the local

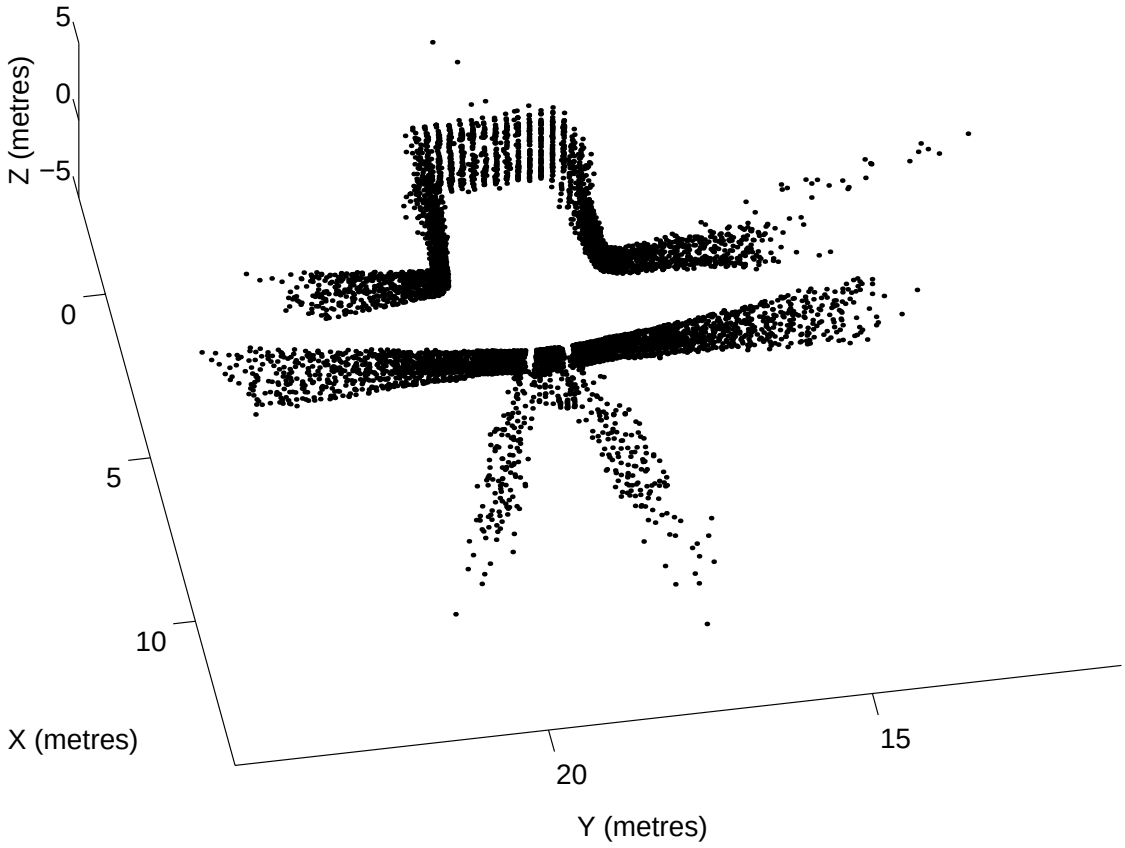

Figure 3: *A 3D range map showing the immediate surroundings of the sensor.*

environment surrounding the sensor (each point shown corresponds to a single range data point).

Another representation of the same scan (this time with a person standing in front of the sensor) is shown in the top two scans in figure 4.

Each pixel in the top scan of figure 4 is illuminated proportional to the *albedo*[2] of each point in the environment. The top row of pixels corresponds to a single 360^o revolution of the sensor's mirror (see the photo in figure 1). The mirror is then automatically tilted downwards about its horizontal axis through 0.5^o and the next row of pixel values is recorded. This process is repeated 70 times to give the full "wrap around geometry" scan shown. The similarity to a black and white photo of the full environment is evident.

[2]which is shown in (M. J. Brownlow 93) to equal the signal amplitude V_r multiplied by the square of the range.

Figure 4: *Each pixel map shows scans with resolution 792 × 70 (ie: a total of 55440 range/intensity values). The top scan shows the albedo image recorded in a corridor and is similar to a black and white photo showing the "wrap around geometry" of the environment. The brighter the pixel, the higher the reflectivity of the scanned point. The second scan shows each pixel value plotted proportional to range. (Darker pixels are closer to the sensor). Notice that the posters on the walls can no longer be distinguished from the walls since they are at the same distance from the sensor. The lower two scans also show albedo and range maps respectively, but were recorded in a more complex environment.*

The second image in figure 4 shows the result of illuminating each pixel proportional to the *range* output generated during the same scan. The darker the pixel, the closer the object is to the sensor. This *range* image is important for AGV navigation, as a robot can immediately see which objects are in its vicinity. Note that one no longer sees the posters on the walls since they are equidistant from the sensor as the walls themselves. This shows the linearity of the sensor as it is able to produce accurate range estimates for almost all reflectivities. The lower two scans in figure 4 show the same type of data but in a more complex environment. Notice that in the albedo scan (third from the top) one can distinguish the bricks from the cement in the wall (left part of the scan) and one can see the poster (and text) on the wall. In the range image (bottom image) the bricks and cement are indistinguishable as is the poster from the wall since they are equidistant from the sensor.

Now that we have briefly explained the sensing modality itself, we proceed by explaining algorithms for processing the data to extract information from 2D and 3D scans in order to localise the mobile robot carrying the sensor.

3 Feature Extraction - Edge Detection

In this section we determine how to segment the range data during a 2D scan, or a 2D section from a 3D scan, which remain after the removal of spurious points (Adams 92; Adams and Probert 96), to find discontinuities in *range* which can be used in mobile robot localisation.

In the literature, (Crowley 85) applies a "least squares" line fit to data from ultrasonic sensors in order to find edges. The method minimises the mean squared error between a set of noisy data and its best line fit, provided data points are used which actually should belong to that line. Line fitting takes place only after a complete 360^o scan has been recorded. Further, all of the recorded range data has equal influence on the equation of the resulting line segment, since no range variance is used when forming the line. When the squared error between a range value and the computed line segment reaches a certain threshold, a new line segment is initiated.

Vision based approaches to edge detection also exist based upon the analysis of complete images (Canny J. 86; Besl P. J. 88).

Instead of attempting to fit straight line segments to sensor data after a full scan has been recorded and then finding the edges, we attempt an alternative approach. As the sensor data arrives, we continuously update a filter which provides a prediction of the coordinates of the next sensor data point, based on previous data, assuming it lies within the same planar surface.

As explained in section 2, we know the approximate range variance associated with each sensed range value. The contribution of each range estimate to the next range prediction is therefore weighted according to its variance. In forming our prediction of 'what range value we next expect' we take more notice of data with small range variance σ_r^2 and less notice of that with higher uncertainty. A *validation gate* equation is then used to decide whether or not the next observed data point is a close enough match to the predicted data point to continue the filter update, and hence the search for an edge. If the observed (sensed) point lies outside of the validation region, a possible edge has been detected and the filter is reinitialised (see (Adams 92; Bozic 86)). The advantage of this technique is that edge detection is an on line process, occurring as the data arrives. This means that a possible edge coordinate is determined immediately after it is sensed. We do not wait for a complete 360^o scan, and then calculate the coordinates of the intersection of line segments to determine edge coordinates.

We now introduce a mathematical framework, namely the *extended Kalman filter* (EKF), for the optimum production of planar edges.

3.1 Edge Detection - The Extended Kalman Filter

Consider the general planar surface shown in figure 5 and the corresponding sensed data points which would result from a perfect 3D line of sight sensor. Application of the sine and cosine

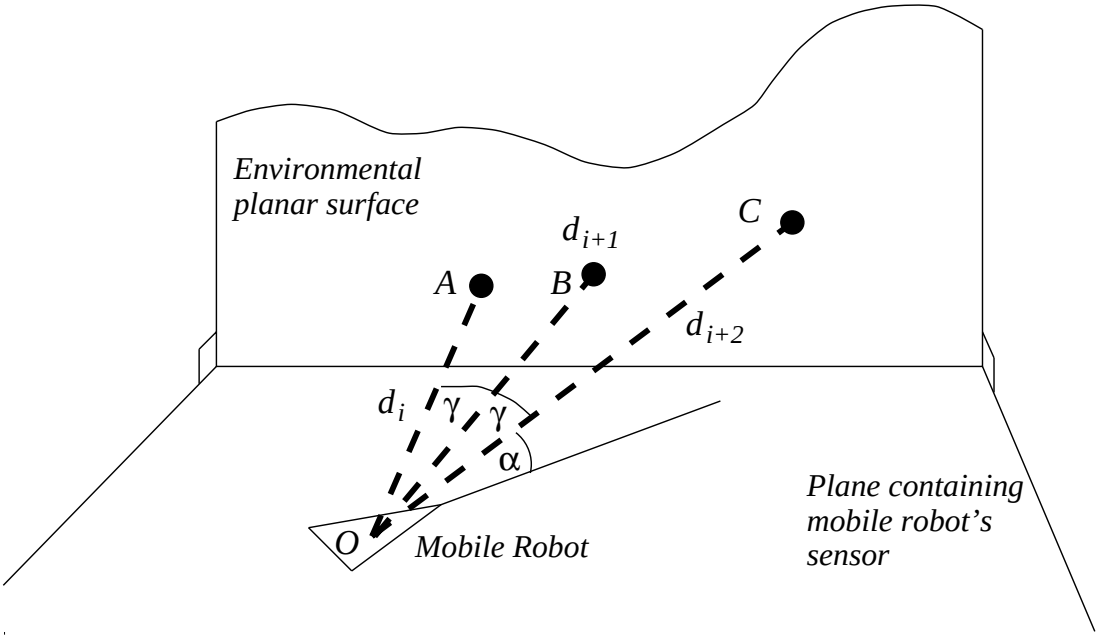

Figure 5: *The relationship between successive range readings when scanning a planar surface.*

rules to the two triangles OAB and OBC in figure 5, and using the fact that the three points A,B and C all lie within a vertical plane, it is possible to show that the relationship between successive range readings, when the light beam is incident upon a planar surface is given by[3]

$$d_{i+2} = \frac{d_i d_{i+1}}{(2d_i \cos\gamma) - d_{i+1}} \tag{4}$$

where γ is the constant angle (in bearing) between successive samples of the sensor as it rotates about its vertical axis. Note that the relationship given in equation 4 is *independent* of the elevation angle α.

3.1.1 System Model

Equation 4 is a second order difference equation with respect to time since range sample d_{i+2} is recorded one time unit after d_{i+1} which in turn is recorded one time unit after d_i. Hence we define the following state space equations:

$$x_1(k+1) = d_{i+2} \tag{5}$$

$$x_2(k+1) = x_1(k) = d_{i+1} \tag{6}$$

where $x_1(k)$ and $x_2(k)$ represent state scalars at discrete time k. Therefore equation 4 can be fully defined by the state space equations:

$$x_1(k+1) = \frac{x_1(k)x_2(k)}{(2x_2(k)\cos\gamma) - x_1(k)} \tag{7}$$

[3]Note that when the light beam rotates in bearing at a constant azimuth angle α, its intersection with a planar surface does *not* in general form a straight line, except for the particular case when $\alpha = 0^o$.

$$x_2(k+1) = x_1(k) \tag{8}$$

In vector form we have that $\mathbf{x}(k+1) = (x_1(k+1) \;\; x_2(k+1))^T$ is a *non-linear* function of $\mathbf{x}(k)$, ie:

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k)) \tag{9}$$

Equation 9 represents a simple system model which will be used to predict the next range value from the sensor (provided a planar surface is being scanned) before the actual range value is recorded. Note that equation 9 is a particular case of the standard system model equation:

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k) \tag{10}$$

but in this particular case we have no input vector $\mathbf{u}(k)$ and if we assume our indoor environment to consist of perfectly planar surfaces (eg. cabinets, open doors, walls) then equation 7 is in fact a perfect description of the state evolution, so that the system noise $\mathbf{v}(k) = \mathbf{0}$.[4]

3.1.2 Observation Model

We now consider the effect of the sensor measurement or *observation* of the states $x_1(k)$ and $x_2(k)$. We note that with any optical range finder, the range variance σ_r^2 can vary considerably depending on the received signal amplitude. For example equation 2 shows that the standard deviation σ_r can be as high as 27 cm for the smallest detectable non zero signal strength of 0.1 volts. For an intermediate signal strength of 1.5 volts, $\sigma_r \approx 2$ cm. For a Gaussian range distribution, $99\frac{3}{4}\%$ of all detected ranges will lie between $\mu - 3\sigma_r$ and $\mu + 3\sigma_r$, where μ is the mean range estimate (Kreysig 83). Hence this boundary corresponds to an uncertainty in range of ± 6 cm for a returned signal with an intermediate (1.5 volts) amplitude.

Due to the small optical beam diameter (1 cm) however, the absolute maximum circumferential bounds upon the range estimate is only ± 0.5 cm. Hence the angular uncertainty in a range estimate is very much smaller than the radial uncertainty. We therefore ignore this small angular uncertainty and consider the range measurement as simply d_{i+1} (d_{i+2} being the next predicted estimate along a planar surface) and assume that its bearing is known precisely. Therefore our observation is simply:

$$z(k) = d_{i+1} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + w(k) \tag{11}$$

where $z(k)$ is the observation and $w(k)$ is a zero mean additive noise process with a known variance σ_r^2. We assume white (or uncorrelated) noise, since our range samples are spaced $n\Delta T$ apart in time, where $n\Delta T >> T_f$, the optical sensor's cut off time (Adams 92). Hence the range samples can be considered to be temporally uncorrelated. We can rewrite the observation equation as:

$$z(k) = \mathbf{C}\mathbf{x}(k) + w(k) \tag{12}$$

where $\mathbf{C}$ is the observation matrix. To find a good or 'best fit' edge, we require that an observation lies outside of some validation region centred on the prediction. To form this prediction and validation region we use the extended Kalman filter algorithm which is described in the appendix.

[4] An interesting future research directive would be to include a finite value for $\mathbf{v}(k)$ to allow for imperfect planar surfaces - ie: surfaces which are slightly curved or contain small irregularities (door handles, pipes on the wall etc.).

4 Edge Detection: Results

In order to demonstrate the edge detection method, we will firstly show an in depth study using 2D scans.

Because the value of γ in equation 4 is assumed to be constant, the extended Kalman filter cycle expects new observations every time the sensor rotates through this angle. The discontinuity (mixed pixel) detector in (Adams and Probert 96) will have labeled some of these observations as "bad data". To continue the Kalman cycle successfully, we replace such 'bad observations' with their predictions, calculated in step 2 of the filter cycle (see appendix), since we can place no confidence in these observations.

The normalised innovation used in the validation gate gives an estimate of the *number of standard deviations* by which the observation differs from its prediction at a given time step, k. Consider the effect of choosing a "three sigma limit" for the validation gate (ie: $D(k+1) = 9$). The normalised innovation is a χ^2 random variable, and tables show that we would be 99.7% certain that the observation lies within $\pm 3\sqrt{s(k+1)}$ (where s is the variance of the innovation - see equation 28 in the appendix) of the position predicted under the assumption that it lies on the same planar surface as its predecessors (Bar-Shalom and Fortmann 88). Two effects can yield a value for $D(k+1)$ greater than 9. *Either* the observation does not lie within the same planar surface as its predecessors, *or* the observation should belong to the same planar surface but happens to lie outside of the "three sigma limit". The latter of these will occur about once in every 400 trials.

Edge detection is very dependent upon the value of the variance associated with a given range estimate. From equation 2 we see that the detection of an edge will therefore depend upon V_r, the received signal amplitude. For an intermediate signal strength of $V_r = 1.5$ volts, the filter will not detect an edge when an observation lies less than approximately 6 cm from its prediction. As would be expected, the filter becomes "more lenient" with data having low values of V_r (high values of σ_r). For example when $V_r = 0.5$ volts, an edge will not be detected only if the observation differs from its prediction by less than 16 cm.

The scans in figure 6 show the technique using range data recorded in a simple laboratory environment. In the scans, the triangle represents the robot carrying the sensor, the simple line model represents a plan view of the environment, and each small cross a single range data point. In the left hand scan, once a value for the normalised innovation, $D(k+1)$ of 9 or more was estimated, a possible edge was detected (shown as the large crosses (+) in the left hand scan) and the filter was reinitialised, in order to begin tracking the next line segment.

Note that edges have been detected at points A and B actually corresponding to a radiator and a vertical pipe, not shown in the simple line map. Although the filter is correct, and shows high sensitivity in detecting these points, the changes in depth which cause these edges are often in practice to small to be of use as features, since it is unlikely they will be found from other mobile robot positions (a factor which is necessary for mobile robot localisation in section 5). Note that using a single threshold value for $D(k+1)$ also results in many false alarms between C and D in figure 6.

In practice we were able to decrease the sensitivity of our edge detector (so that only larger range discontinuities would be detected) by using two *different* threshold values for the normalised innovation $D(k+1)$, one to indicate a possible edge which we will call g, and another lower value which is just used to reinitialise the filter which we will call h. If we choose h to be unity, then any observation differing from its prediction by more than $\pm\sqrt{s(k+1)}$ (see equation 28 in the appendix) will reinitialise the filter so that it tracks a new line segment. This results in better predictions, since the small range changes which now reinitialise the filter (but are not considered large enough to indicate an edge), previously caused bad predictions to be made,

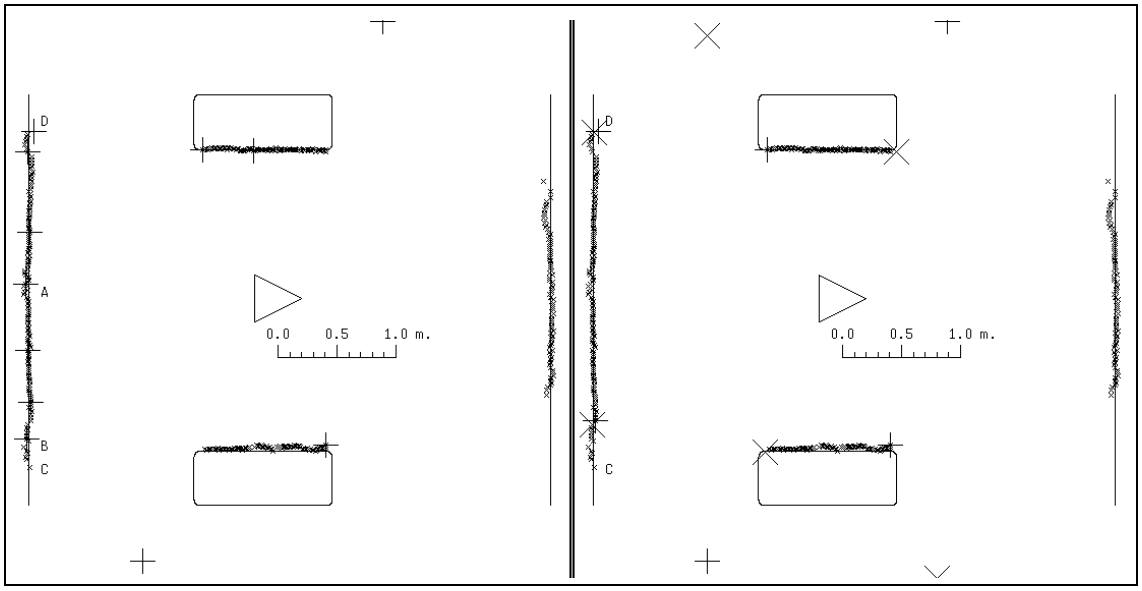

Figure 6: *Kalman edge detection, using data which has passed through the mixed point detector (Adams and Probert 96) successfully. The left hand scan shows crosses (+) at detected edges when a '3 sigma limit' is assigned to* $D(k+1)$*. The right scan shows the result of splitting* $D(k+1)$ *into two thresholds,* $g = 9$ *and* $h = 1$ *(see text).*

eventually leading to the detection of edges such as those at points A and B.

The crosses (+) in the left hand scan of figure 6 occur when the observation significantly differs from the prediction meaning that the previous observation was the last point on the previous line segment. Hence these points should also be marked as possible edges, and these are shown as diagonal crosses ($\times$) in the right hand scan in figure 6.

The effect of setting the normalised innovation $D(k+1)$ threshold to a single value of 9 is demonstrated by the left hand graphs in figure 7. The top left hand graph shows the observed ranges (solid line) and the predicted ranges (dashed line) along the wall CD in figure 6. The small range changes at points A and B (these points also corresponding to A and B in figure 6) cause erratic predictions which then result in the filter indicating an edge. The left hand lower graph in figure 7 indicates the variation of $D(k+1)$, the normalised innovation, with sensor azimuth between 140^o and 220^o. A logarithmic scale is used (the vertical axis is $\log_{10}(\sqrt{D(k+1)})$) because of the vast range of values of $D(k+1)$. For example, between C and D, $D(k+1)$ was mostly less than unity, but at C it rose to over 120. The horizontal dashed line shows the chosen threshold ($D(k+1) = 9$) beyond which the filter indicates an edge.

The right hand scan in figure 6 shows the edge detector running on the same data, this time with the threshold h reduced to unity, g remaining at 9. Note that the filter still detects the 'important' edges at the corners of the pillars, but is now less sensitive to the small range changes along wall CD as it now only detects one edge shown near to point C. The right hand graphs in figure 7 show that the predictions are a closer match to the observations, particularly at points A and B. The reason for this is shown in the lower right hand graph. When the curve rises above $\log_{10}(d) = 0$, $D(k+1) > 1$ and the filter is reinitialised. This occurs at A and B (and at various other points) so that the predictions at these points are based on a new line segment. The dashed line again shows the threshold value beyond which the range points are marked as starting a new segment and an edge is assumed. Note the degraded sensitivity, since only 2 points now rise above this threshold.

The scan in figure 8 shows the edge detecting algorithm running on data which has suc-

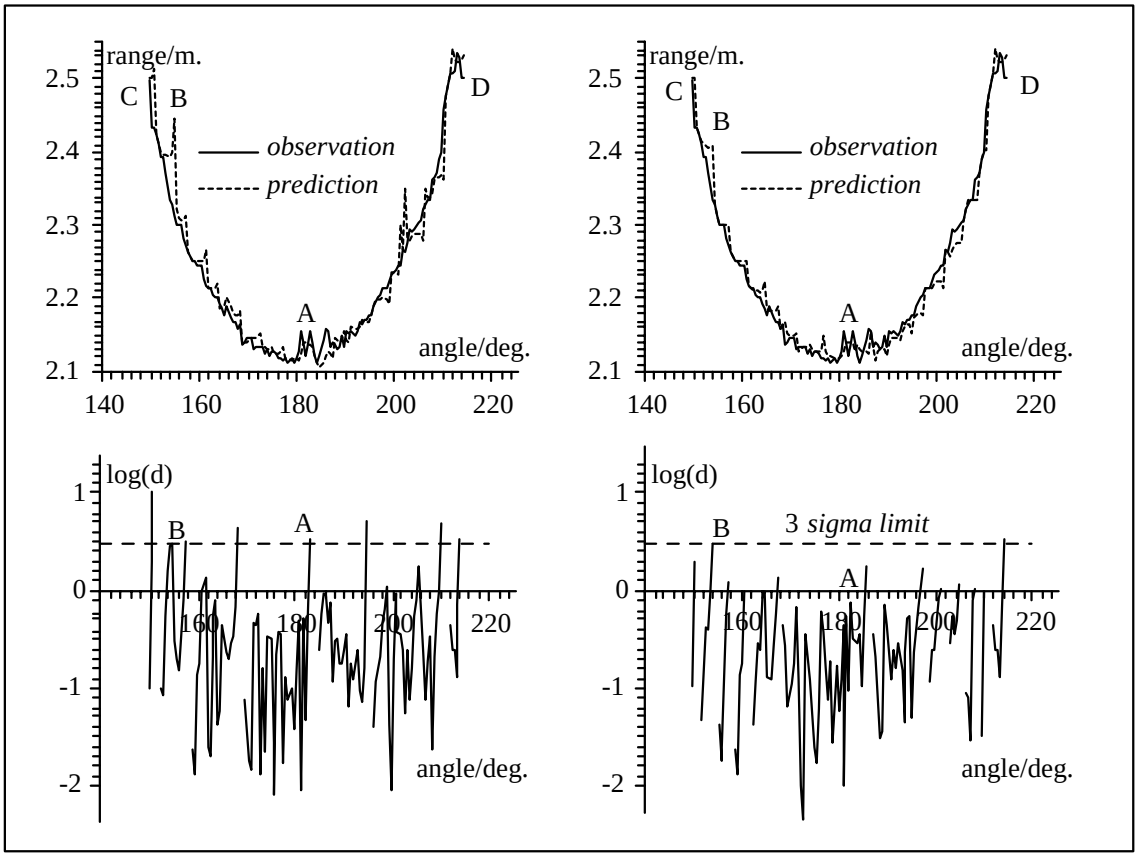

Figure 7: *Graphs of range prediction and observation versus sensor azimuth along the wall marked CD (top graphs). The left hand curves are the results when the threshold value for* $D(k+1)$ *has a single value* ($g = h = 9$). *The right hand curves show the closer match between prediction and observation when* $g = 9$ *and* $h = 1$. *The lower graphs show* $\log_{10}(\sqrt{D(k+1)})$ *for each observed data point.*

cessfully passed through the discontinuity detector given in (Adams and Probert 96), recorded within the same environment, but from a different position. Once again the two thresholds for $D(k+1)$ were separated so that $g = 9$ and $h = 1$. The large range discontinuities estimated by the filter are marked at the pillar edges where values of the normalised innovation $D(k+1)$ rose above 9. These points may be interpreted as range discontinuities which are large enough to represent the edges of obstacles.

The graphs in figure 9 show various parameters produced in the filter cycle, using only the data from the right hand pillar's lower detected edge in figure 8. The top graph shows the observed and predicted range estimates just before this edge is detected. Superimposed upon these curves are curves of $\hat{x}_1(k+1 \mid k) \pm 3\sqrt{s(k+1)}$, labeled as the '3 sigma limit' and $\hat{x}_1(k+1 \mid k) \pm \sqrt{s(k+1)}$, labeled as the '1 sigma limit', where $\hat{x}_1(k+1 \mid k)$ is the first component of the range prediction (equation 19 in the appendix). The 3 sigma limit indicates the region within which the point is assumed collinear, but if an observation moves outside the 1 sigma limit, the filter is reinitialised. The latter occurs at point N in the top graph. It can be seen that the corresponding point on the lower graph is the first to rise above 0, meaning that $D(k+1) > 1$. The next two observations are used to initialise the filter (Adams 92). At these two points the predictions equal the observations and no values can be computed for $s(k+1)$ or $D(k+1)$ until the third observation is recorded. This explains the gap in the graphs of figure 9.

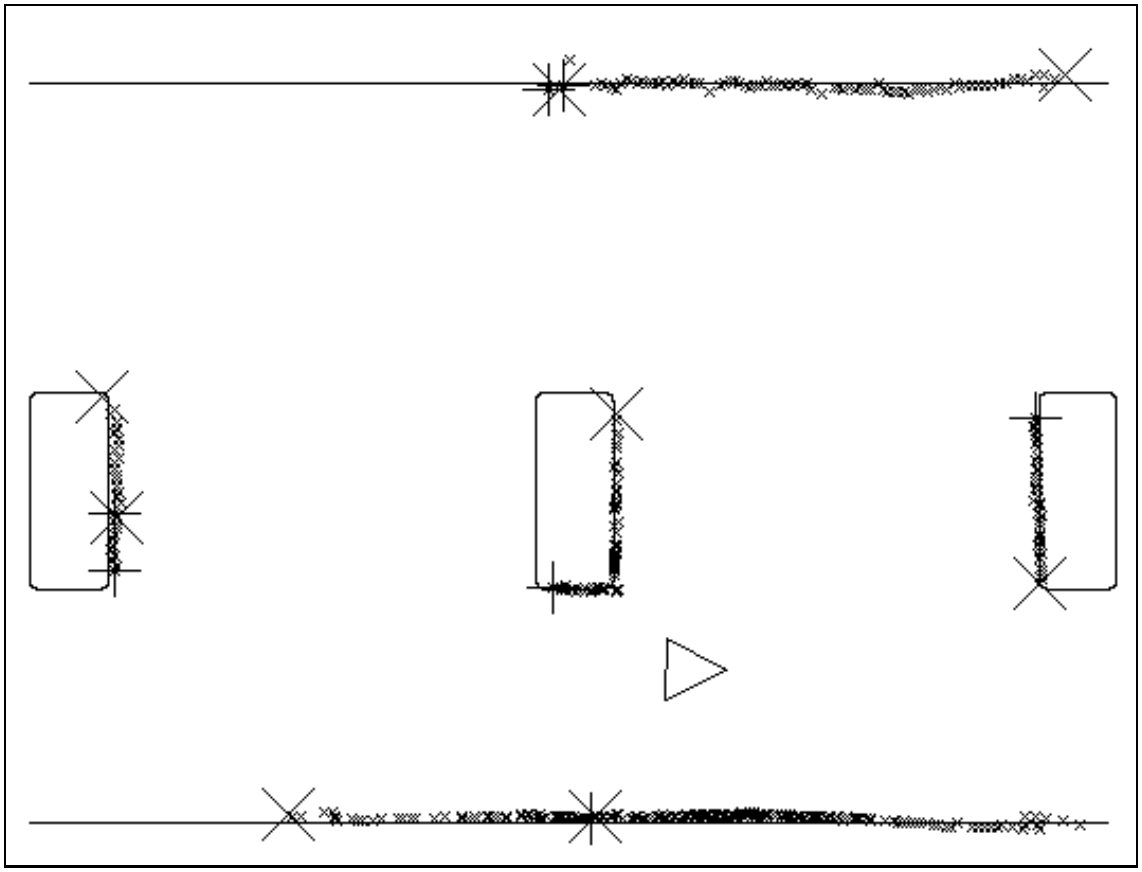

Figure 8: *The edge detector produces the edges shown when the threshold values for $D(k+1)$ are split such that $g = 9$ and $h = 1$. Edges are shown by crosses: $\times$ when the range is increasing as the sensor rotates clockwise and $+$ when the range is decreasing.*

Point M shows the observation curve cutting the 3 sigma envelope as the next observation lies outside of this limit. Hence the filter estimates the previous observation as the end point of the last line segment, and the current observation as the starting point of a new line segment.

Figure 10 shows the results of the edge detection algorithm in a more cluttered environment. A clear comparison can be seen between detected edges and true edges. The dotted lines represent the actual environment, the cross ($\times$) represents the hand measured position of the mobile robot and the continuous line represents the actual sensor data (one point every 1.3^o). The small stars ($*$) correspond to edges detected within the scan (see section 3.1).

The edges found in this cycle are stored and forwarded to the mobile robot localisation algorithm which will be discussed in the next section.

5 Mobile Robot Localisation

Localisation is the process of determining the position of the mobile robot with respect to a global reference frame (Leonard and Durrant-Whyte 90). Now that we have a reliable algorithm for the extraction of particular features during single scans, we proceed with our localisation algorithm. We present here a mathematical model which statistically links the positions of these detected features (relative to the robot), the uncertainty in the range readings used to find these features, and a simple model of the motion of a mobile robot in order to form an optimal estimate of the robot's position.

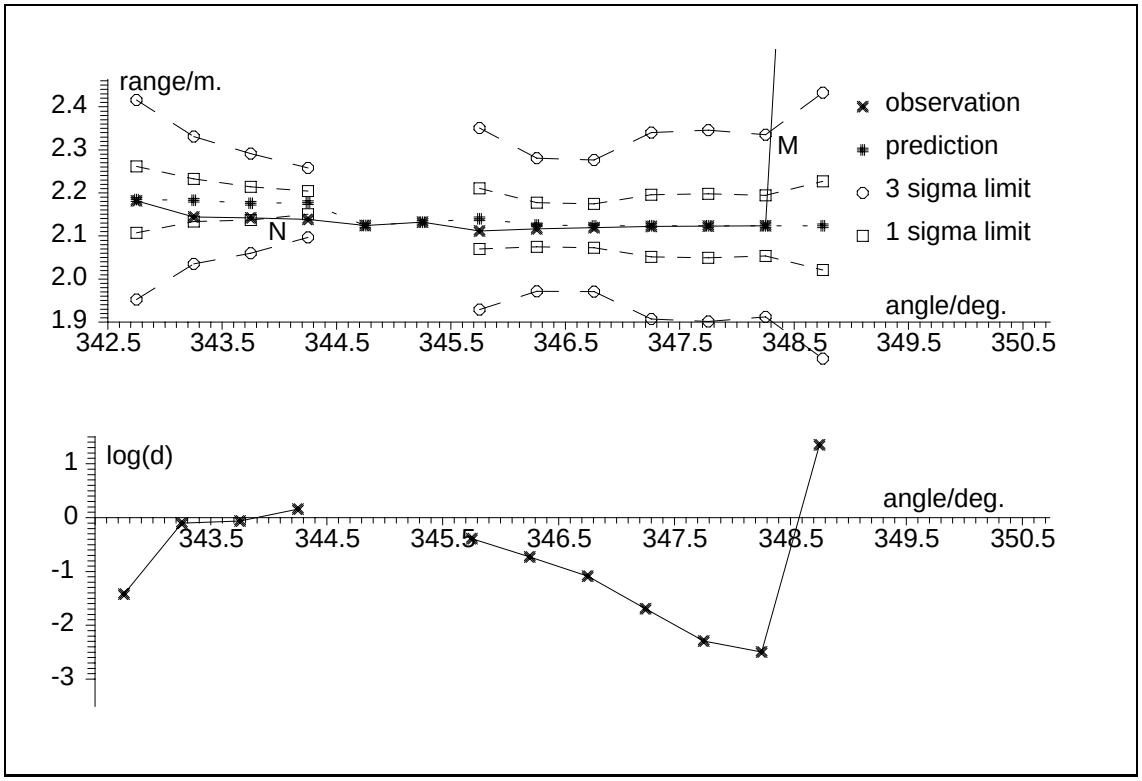

Figure 9: *The top graphs show the observations, predictions, "3 sigma limits" and the "sigma limits" in the vicinity of the right hand pillar's detected edge in figure 8. The lower graph shows the value of* $\log(d) = \log_{10}(\sqrt{D(k+1)})$.

5.1 Mobile robot kinematics - The system model

The system model describes how the vehicle's position $\mathbf{x}(k)$ changes with time in response to a control input $\mathbf{u}(k)$ and a noise disturbance $\mathbf{v}(k)$. It has the form:

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k) \tag{13}$$

where $\mathbf{F}(\mathbf{x}(k), \mathbf{u}(k))$ is the (non-linear) state transition function. $\mathbf{v}(k)$ is a noise source assumed to be zero-mean Gaussian with covariance $\mathbf{Q}(k)$. The control input is $\mathbf{u}(k) = [T(k), \Delta\theta(k)]^T$. This model is based on point kinematics and consists of a rotation clockwise through the angle $\Delta\theta(k)$, followed by a translation forward through the distance $T(k)$ in the plane of the mobile robot (figure 11). The translation function then has the form:

$$\mathbf{F}(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} \theta(k) + \Delta\theta(k) \\ x(k) + T(k)\cos\theta(k) \\ y(k) + T(k)\sin\theta(k) \end{bmatrix} \tag{14}$$

5.2 2D Scanning: The Observation Model

After the process of feature extraction from the raw sensor data, we arrive at a set of observations

$$Z(k) = \{\mathbf{z}_j(k) \mid 1 <= j <= n_o\}. \tag{15}$$

The observation model describes how the measurements $\mathbf{z}_j(k)$ are related to the vehicle's position and has the form:

$$\mathbf{z}_j(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{p}_t) + \mathbf{w}_j(k), \quad \mathbf{w}_j(k) \approx \mathbf{N}(\mathbf{0}, \mathbf{R}_j(k)). \tag{16}$$

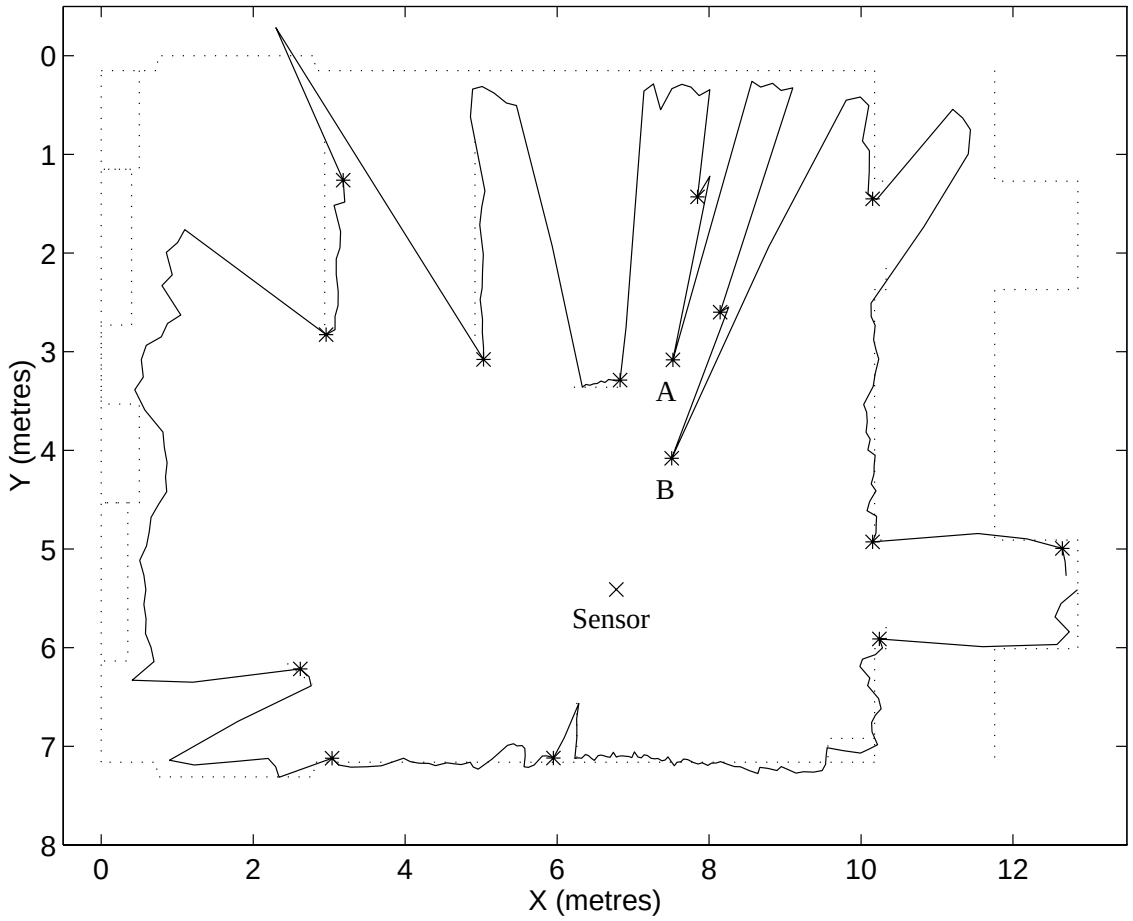

Figure 10: *Edge detection with the EKF in a more cluttered environment. The dotted lines represent the actual environment, the cross (×) represents the hand measured position of the mobile robot and the continuous line represents the actual sensor data (one point every 1.3º). The small stars (∗) correspond to edges detected within the scan.*

In the 2D case, the target state vector has the form $\mathbf{p}_t = (p_x, p_y)$, (ie: the x, y coordinates of an edge) p_z being 0 in this case. The measurement function $\mathbf{h}(\mathbf{x}(k), \mathbf{p}_t)$ expresses an observation $\mathbf{z}(k)$ from the sensor to the target as a function of the robot's location $\mathbf{x}(k)$ and the target coordinates $\mathbf{p}_t$. In the case of a *two dimensional* scan (in a two dimensional plane) we see that:

$$\mathbf{h}(\mathbf{x}(k), \mathbf{p}_t) = \begin{pmatrix} \text{Range} \\ \text{Bearing} \end{pmatrix} = \begin{bmatrix} \sqrt{(p_x - x(k))^2 + (p_y - y(k))^2} \\ \tan^{-1}\left(\frac{P_y - y(k)}{P_x - x(k)}\right) - \theta(k) \end{bmatrix} \tag{17}$$

Each measurement is assumed corrupted by a zero-mean, Gaussian noise disturbance $\mathbf{w}_j(k)$ with covariance $\mathbf{R}_j(k)$.

5.3 3D Scanning: The Observation Model

In the case of 3D scanning, the system model in equation 14 remains unchanged since the robot moves in a 2D plane. The observation model changes however, as we now have a different relationship between our point observation coordinates (p_x, p_y, p_z) and the robot's state (position).

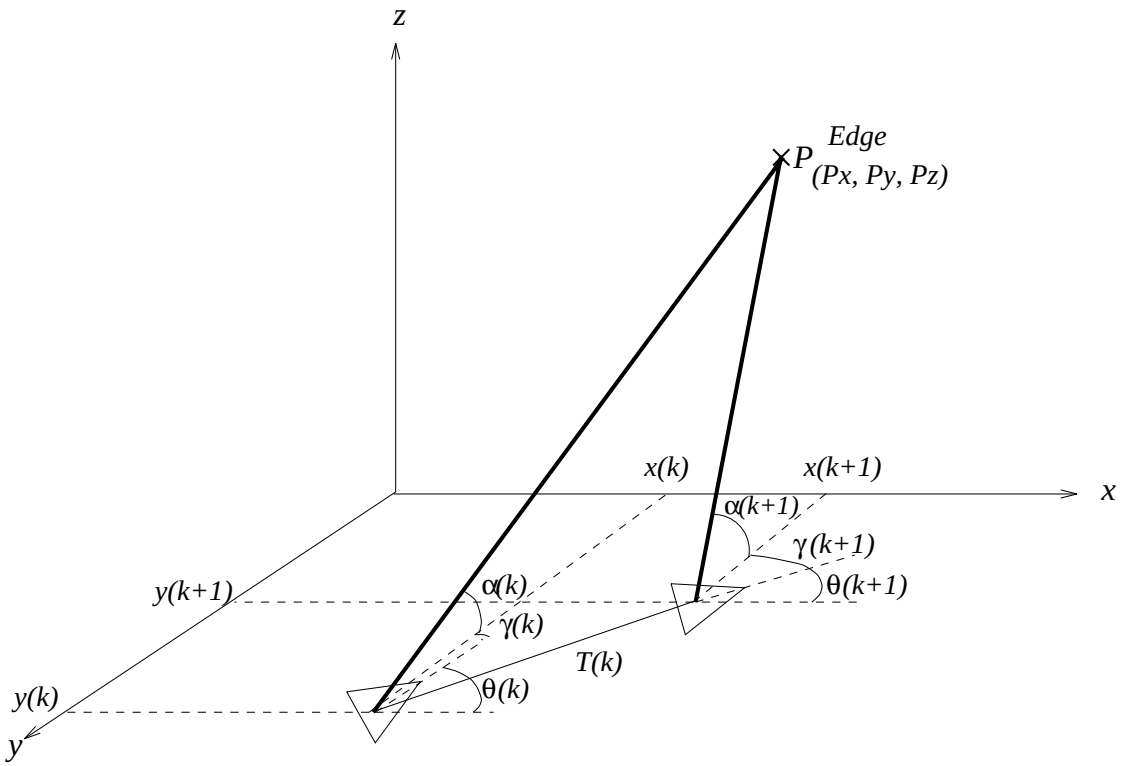

Figure 11: *Localisation by tracking a feature in 3D space.*

In this case the non-linear observation matrix $\mathbf{h}(\mathbf{x}(k), \mathbf{p}_t)$ in equation 16 becomes (see figure 11):

$$\mathbf{h}(\mathbf{x}(k), \mathbf{p}_t) = \begin{pmatrix} \text{Range} \\ \text{Bearing} \\ \text{Azimuth} \end{pmatrix} = \begin{bmatrix} \sqrt{(p_x - x(k))^2 + (p_y - y(k))^2 + p_z^2} \\ \tan^{-1}\left(\frac{P_y - y(k)}{P_x - x(k)}\right) - \theta(k) \\ \tan^{-1}\left(\frac{P_z}{[(P_x - x(k))^2 + (P_y - y(k))^2]^{\frac{1}{2}}}\right) \end{bmatrix} \tag{18}$$

5.4 The Localisation Cycle

Now that we have defined our mobile robot kinematic model and sensor to target observation equations, we proceed to explain the localisation cycle. The goal of this cycle is to find the new *a posteriori* position estimate $\hat{\mathbf{x}}(k+1 \mid k+1)$ and its covariance $\mathbf{P}(k+1 \mid k+1)$, given the *a priori* vehicle position estimate $\hat{\mathbf{x}}(k \mid k)$ and its covariance $\mathbf{P}(k \mid k)$ at time k, the current control input $\mathbf{u}(k)$ and the current set of observations $Z(k+1)$.

Each separate step in the cycle is discussed in the appendix.

6 Localisation results

We firstly present the localisation results from 2D scans (where the mirror is constrained to scan within a single plane i.e. in bearing only). All the 2D scans are recorded in a laboratory environment.

Two methods for localisation have been used. The first one makes use of a map of the environment. This map consists of a list of edge coordinates in a global reference frame, relative to which the mobile platform can match the edges extracted from a scan. The second method

does not make use of a map, but uses prior scans as a reference for the matching process. Edges found in prior scans are used for matching with edges extracted from the latest scan.

6.1 Localisation *with* an a priori Map

Figure 12 shows a single scan and position of the mobile robot during a run. The symbols used

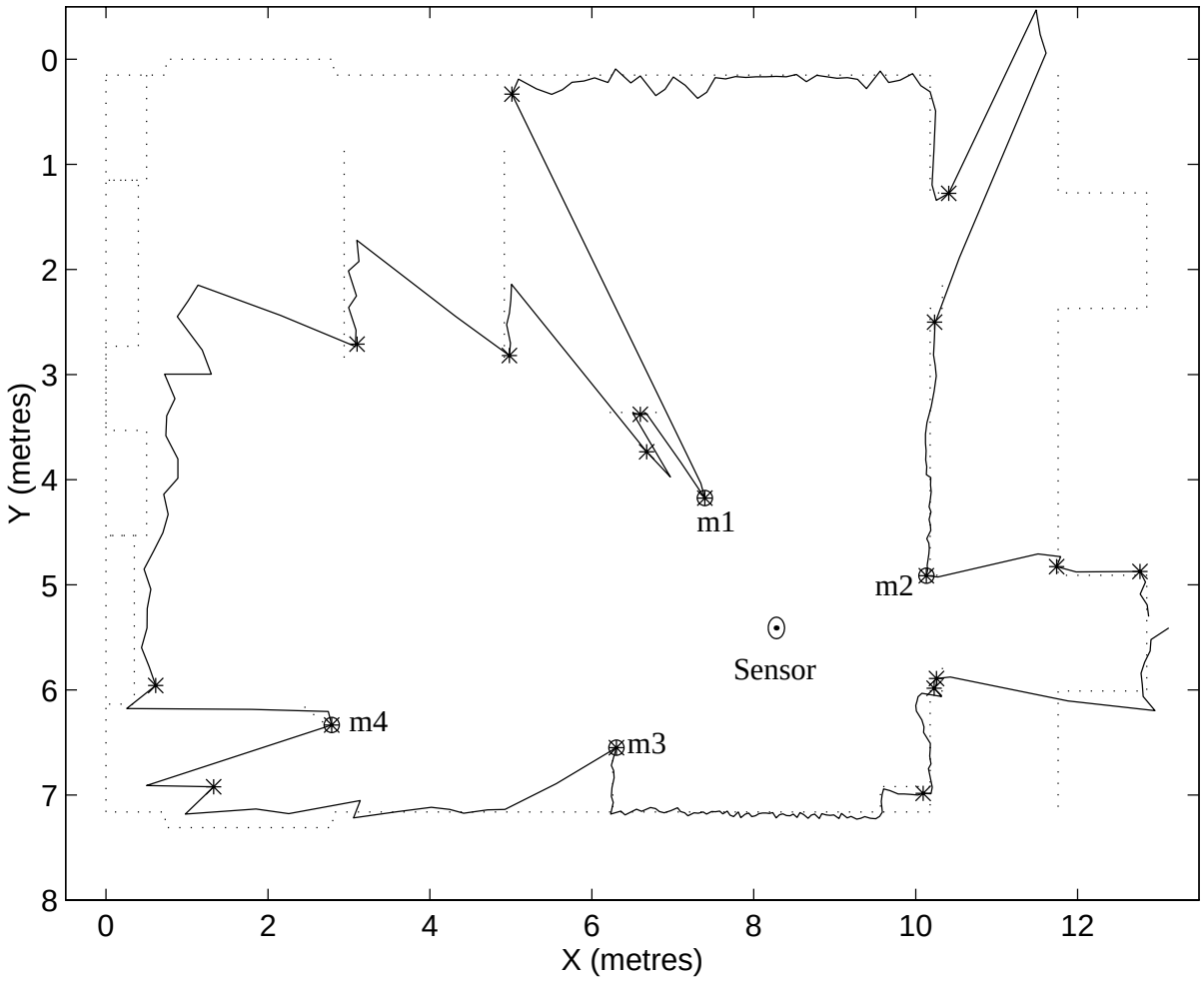

Figure 12: *A single scan and position update during a mobile robot run. The circles (∘) are edges which have been matched to a predicted edge from an edge list in the a priori map. The estimated position of the robot is shown by the point (·) and centred at this estimate is the positional uncertainty ellipse.*

in this figure are the same as those used in figure 10. The circles (∘) are edges which have been found during this particular scan and have been matched, using equation 31 in the appendix, to a predicted edge from an edge list in the *a priori* map. These matched edges have been labeled m1, m2, m3 and m4. The estimated position of the robot is shown by the point (·) and centred at this estimate is the uncertainty ellipse which has major and minor axes equal in length to the square roots of the eigenvalues of the x, y component matrix of the error covariance matrix $\mathbf{P}$ (equation 38 in the appendix) in the direction of the orthogonal eigenvectors of this matrix (Leonard and Durrant-Whyte 90).

Figure 13 shows a new scan after the robot has moved 50 cm to the left. The symbols used are the same as those in figure 12. Note that in this scan four edges have also been matched

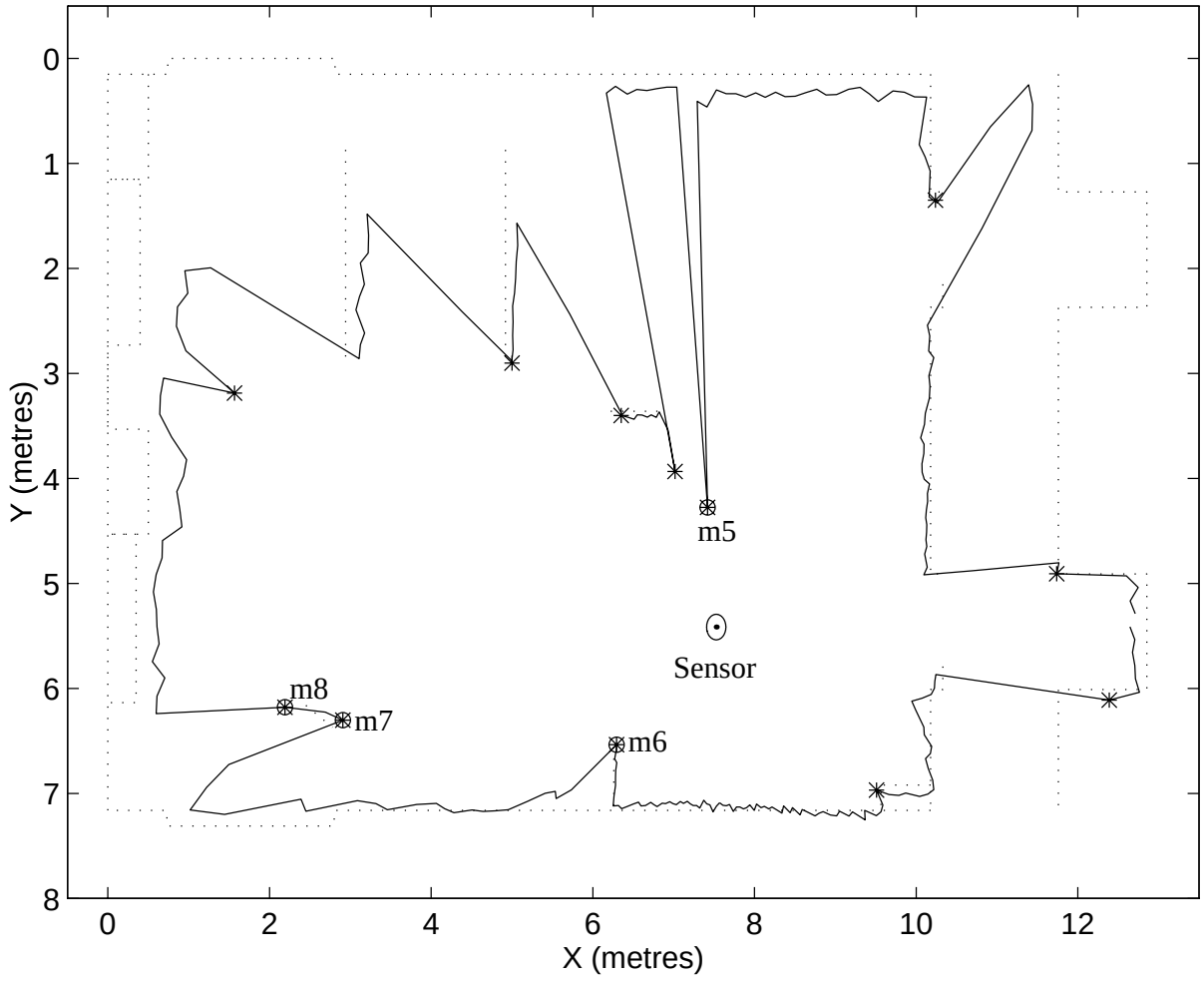

Figure 13: *A 2D scan and position update after the mobile robot has moved 50cm in the negative x direction.*

(m5, m6, m7 and m8) and used to form the predicted position of the mobile robot (marked as the point $(\cdot)$) and its error covariance ellipse. The ellipsi in figures 12 and 13 correspond to regions of minimum area in the xy plane which are known to contain the robot with a constant probability (in our case 95%) (Bar-Shalom and Fortmann 88).

Figure 14 shows the first 8 2D scans, matched edges and positional estimates as the robot traversed its environment. In these scans only *matched* edges are shown (as small circles). Figure 15 shows the next 8 scans during the run.

To demonstrate more clearly the qualities of the localisation algorithm itself, figure 16 shows just the environment and the hand measured, odometric and EKF estimates of the mobile robot's position for all 23 positions recorded during the run. Positions 1 (bottom right corner) to 16 are recorded in a laboratory of approximately 10 x 7 meters (these are the positions shown in figure 14 and 15) after which, the vehicle drove into a small storage room of 3 x 7 meters next to the lab (positions 17 to 23). The crosses $(\times)$ represent the odometric estimates, the dots $(\cdot)$ the EKF estimates, centred on these the uncertainty ellipsi are shown and the triangles

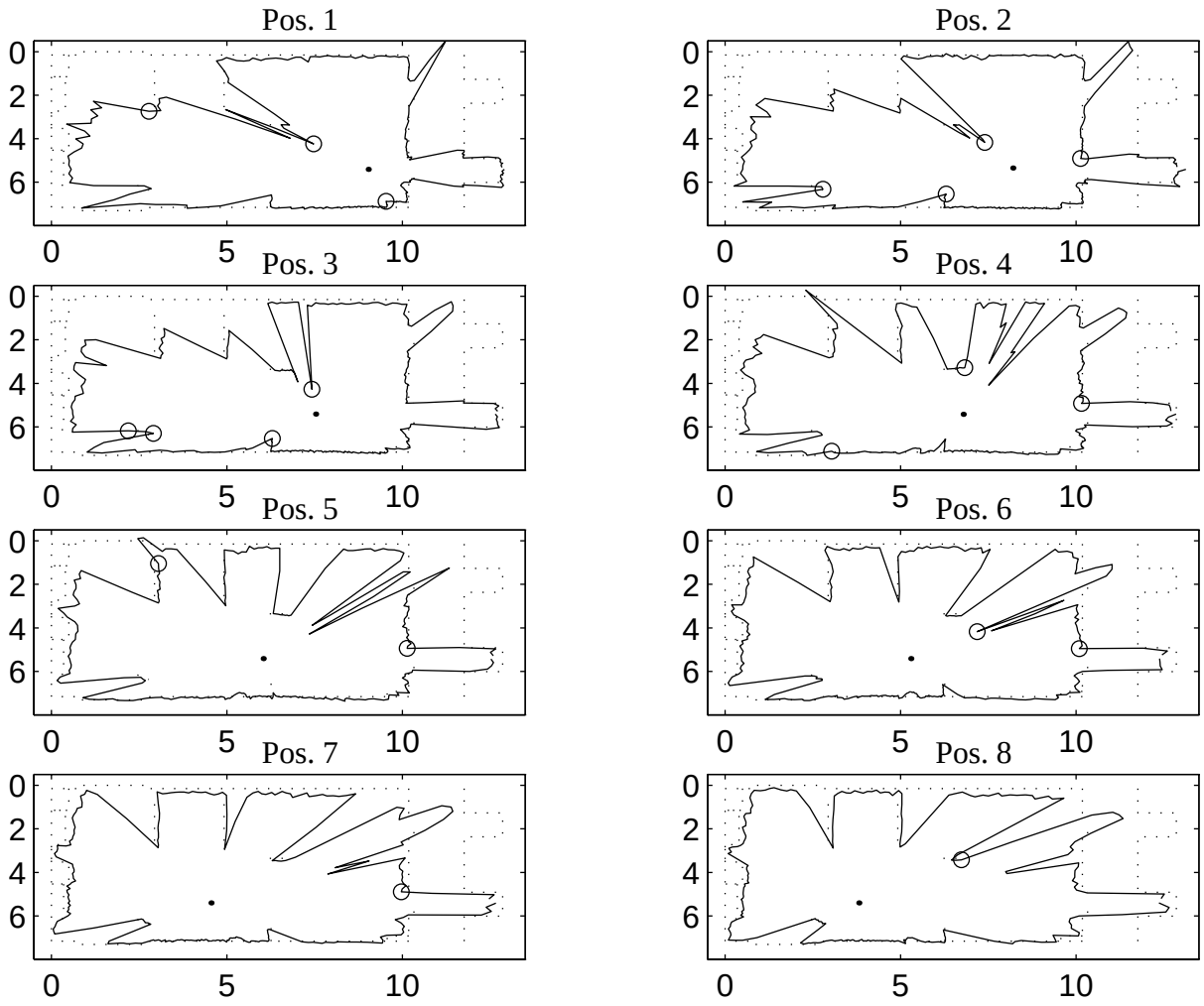

Figure 14: *2D scans, matched edges (circles (∘)) and position updates (dots (·)) as the robot traverses its environment.*

(△) represent the hand measured positions of the robot. Note that the odometric positional estimates diverge well away from the actual positions. It was clearly noted that the ellipsi grew and shrunk as less or more features were matched respectively.

Odometry errors are simulated by giving a systematic error and a small random error with every translation.

6.2 Localisation *without* an a priori Map

The overall result of localisation without any prior known map is shown in figure 17. Here, the first robot position is assumed given (shown as a triangle - hand measured position), because the edges extracted from the scan recorded at this position are used as a reference for later recorded scans. The symbols used in figure 17 are the same as those used in figure 16. A dynamic edge list is kept during this run, where edges are added and removed as they enter and leave the field of view of the sensor. In this run, edges are removed from the edge list if they have not been matched five times in a row. New edges are added as they come into view of the sensor. In

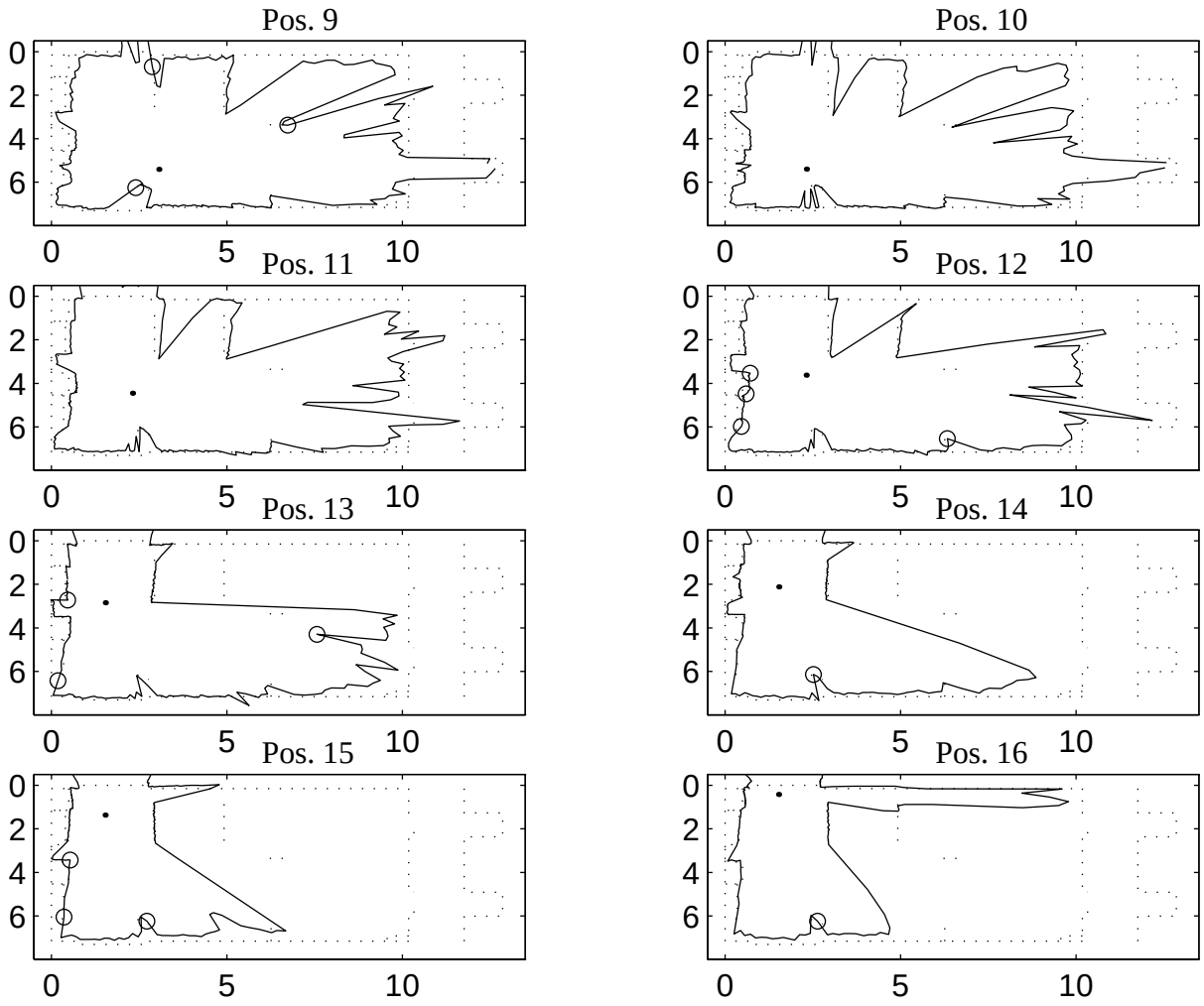

Figure 15: *2D scans, matched edges (circles (∘)) and position updates (dots (·)) as the robot continues to traverses its environment.*

scans 2, 16, 17 and 19 no edges are matched and it was noted that the error covariance matrix eigenvalues increased.

It can be seen that the position as measured by the odometry diverges quickly from the real hand measured position (at the end of this localisation run, the x positional odometric error has increased to 1.5 meters), while the error in the *estimated* x position is almost always less than 0.04 meters. If only one edge in the environment can be matched to the map and the error between this edge and the corresponding edge in the map is large, then the error in the estimated position increases also. The spurious edges discussed earlier in section 2.1.1 which are sometimes undetected, are also a source of problems, because they can introduce large errors in the estimated position of the mobile vehicle (Adams 92). This problem presents itself more clearly in the localisation without any prior map. The chance that spurious edges are matched against each other is much larger in this method than in the map based localisation.

Hence *map based localisation* offers a more robust method for determining the position of a mobile robot, since this method is relatively insensitive to falsely sensed edges. A further point noted by Leonard and Durrant-Whyte, and by Smith, Self and Cheeseman, is that if

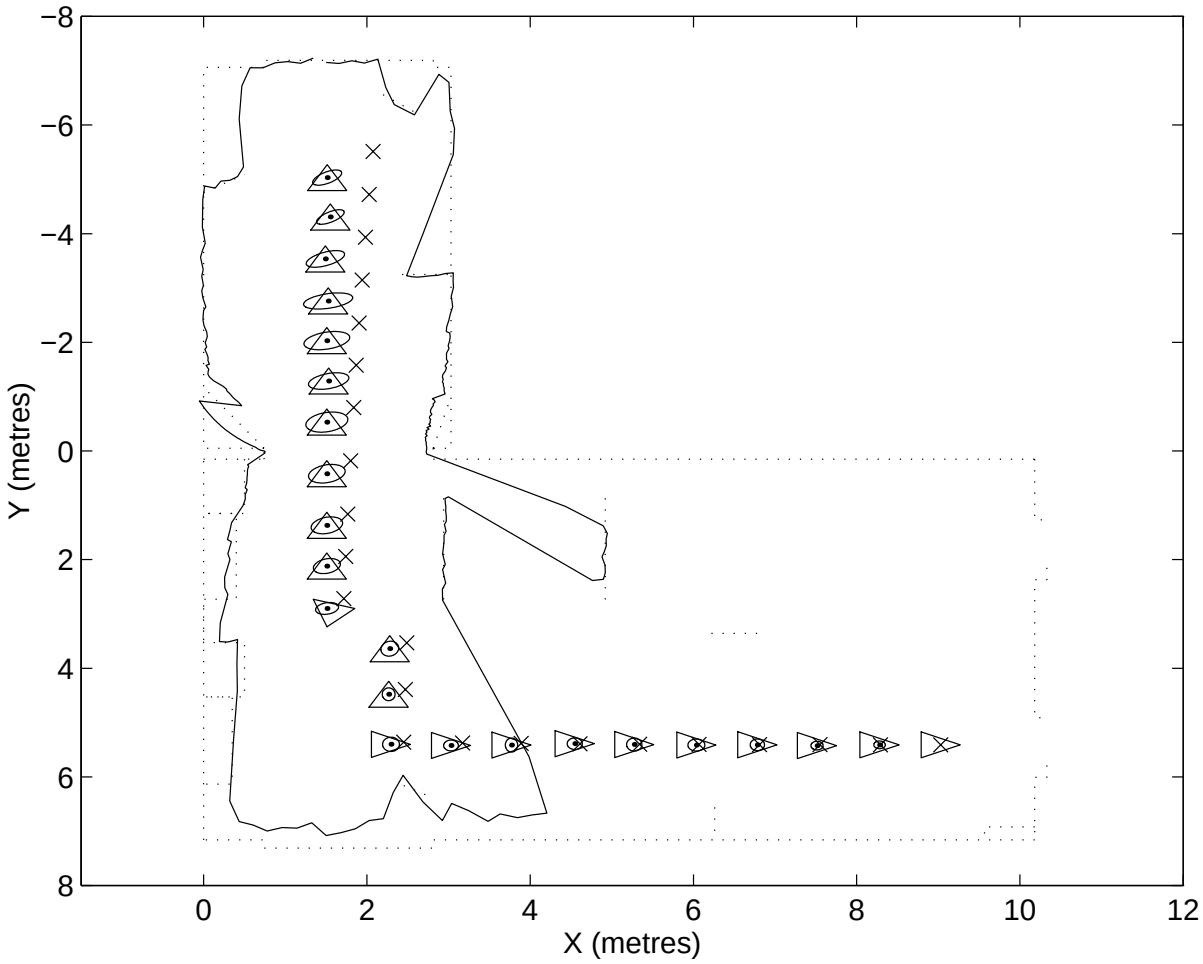

Figure 16: *All 23 positional estimates using the map-based localisation algorithm. The crosses show the odometric estimated positions.*

the observed edges are matched with previously sensed edges, having uncertain positions (i.e. not from a fixed map), then the position estimate produced under the E.K.F. cycle will be correlated with the estimated edge locations (Leonard and Durrant-Whyte 92; Smith et al. 90). This theoretically disobeys the assumptions used in deriving the E.K.F. positional estimates, meaning that an exact solution to the problem should take into account these correlations. Leonard and Durrant-Whyte propose a simple method for side stepping this problem, namely that the robot chooses where it travels and where it focuses the attention of its sensors. This is done to remove correlations. This means that the robot in our case would need to know the positions of some edges precisely.

Hence we have shown in this section, and in figure 17, that the mobile robot localisation algorithm, without an *a priori* map, is capable of producing positional estimates which are more reliable than odometry alone. We acknowledge however that, on a theoretical basis, the robot still requires exactly known landmarks (i.e. some form of simple map) at regular intervals in order to stop it becoming lost. An interesting future research directive indeed remains whereby *either* an exact method which accounts for the cross correlation between uncertain positional information and uncertain edge coordinates is defined (as proposed by Smith *et al* (Smith et al.

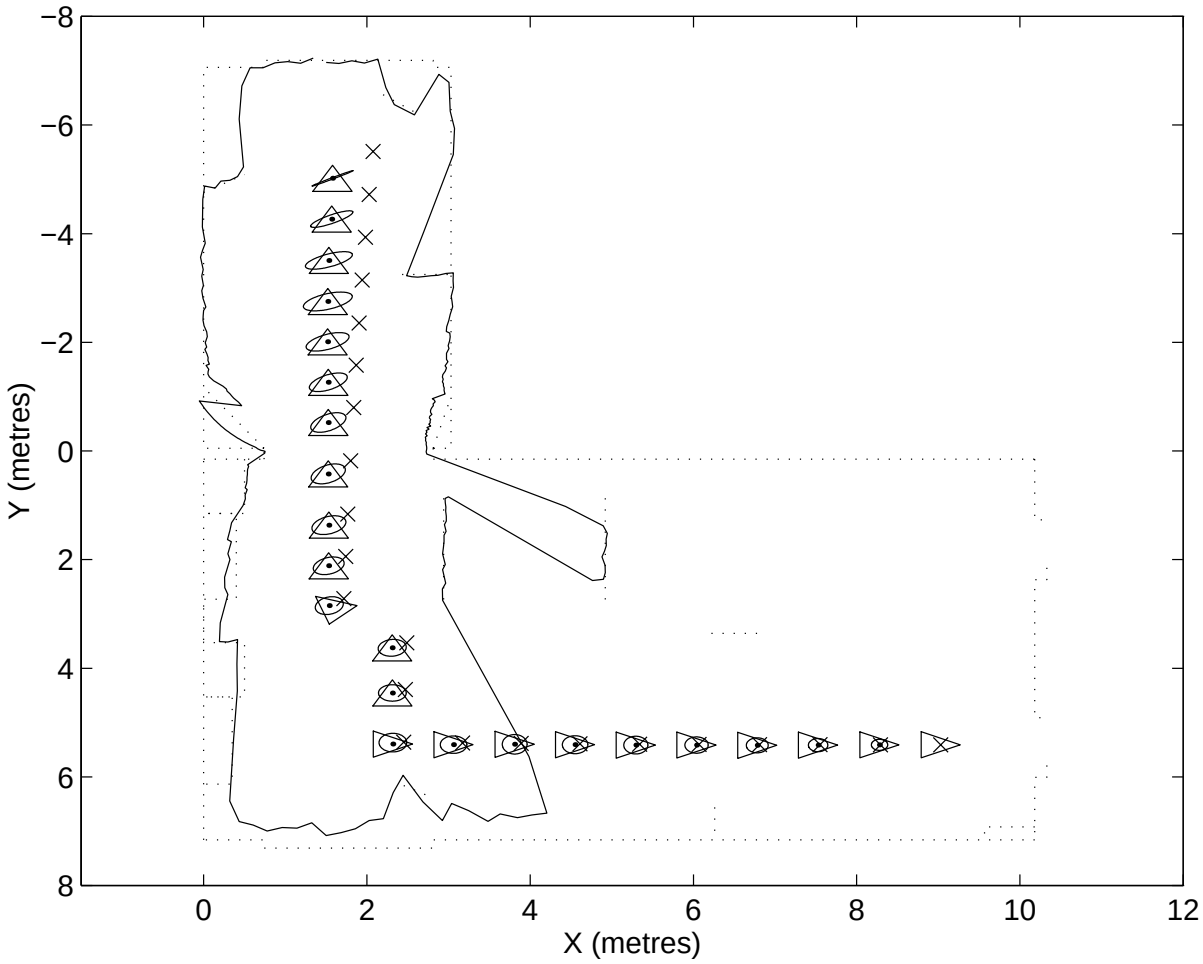

Figure 17: *All 23 positional estimates using the localisation algorithm without any prior known map. Again, the crosses show the odometric estimated positions.*

90)), *or* a somewhat more relaxed method which quantifies how often exact feature information is required in order to provide reliable positional updates for practical mobile robot navigation.

In the next section we will show some results of mobile robot localisation using three dimensional scans of the environment.

7 3D Mobile Robot Localisation

In section 5.3 we derived an observation model to relate a detected edge, which in general lies outside of the plane of the sensor, having coordinates (P_x, P_y, P_z), to the true position of the sensor.

A problem inherent to 3D scanning is that the number of edges detected in a full scan becomes very large. For example figure 12 in section 6.1 shows 18 detected edges in a single 2D scan. From a full 3D scan we can therefore easily expect hundreds of edges to be detected. If a general *a priori* map were then used for matching edges in 3D, it would need to contain an enormous edge list, meaning that computer memory and timing problems could result. For this reason we do *not* use an *a priori* map for 3D localisation but instead use the edges detected in

a restricted number of prior scans as a reference for the matching process. We will show here that reliable edge matching can, in general, only take place between 3D scans which have been recorded in close proximity of each other. This automatically restricts the number of candidate edges which need to be recorded for matching purposes. Also, because of the large amount of edges found in a full 3D scan, we further instruct the algorithm to search for edges within a constrained region, relative to the global coordinate system.

To demonstrate this technique we recorded a scan by rotating the sensor head 30 times (about the vertical axis) in 1.0^o azimuth increments. To high light the detected edges, figure 18 shows a 3D line model of a 210^o section of the mobile robot's environment, the position of the robot and the edges detected between the bearing scanning angles $100^o < \theta < 120^o$ (marked as stars). 15 edges were detected altogether. The scan data itself is not shown so that the detected

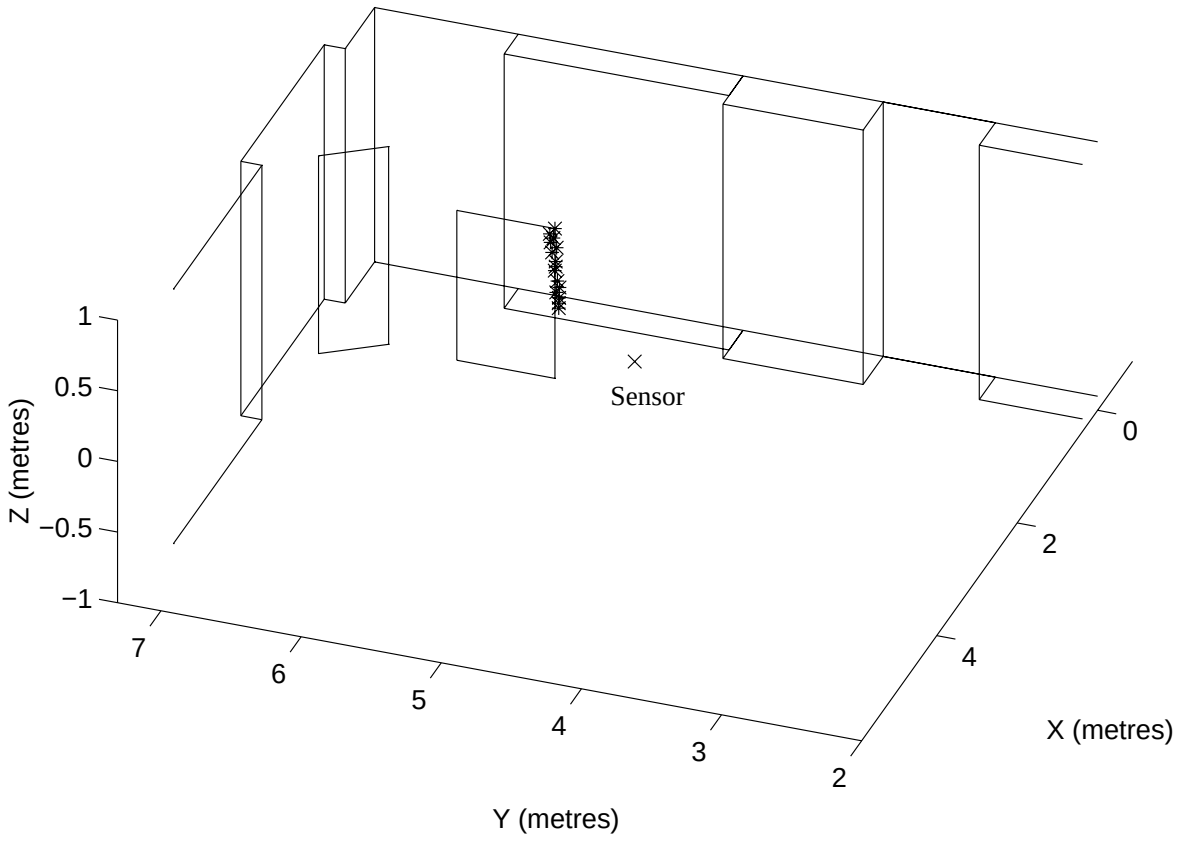

Figure 18: *A 3D line model of a section of the mobile robot's surroundings. The small stars show all edges detected in a small angular region of the environment. The single cross marked 'Sensor' represents the mobile robot 'basis' position.*

edges are clearly visible.

This is considered to be the *basis* or start position where the actual position of the mobile robot is assumed known. The detected edges in figure 18 are used for matching purposes in later scans.

Figure 19 shows 3D scan data recorded 0.8m in the y direction from the 'basis' scan point of figure 18. For clarity, the scan is shown for bearing angles $40^o < \theta < 250^o$ only. This scan

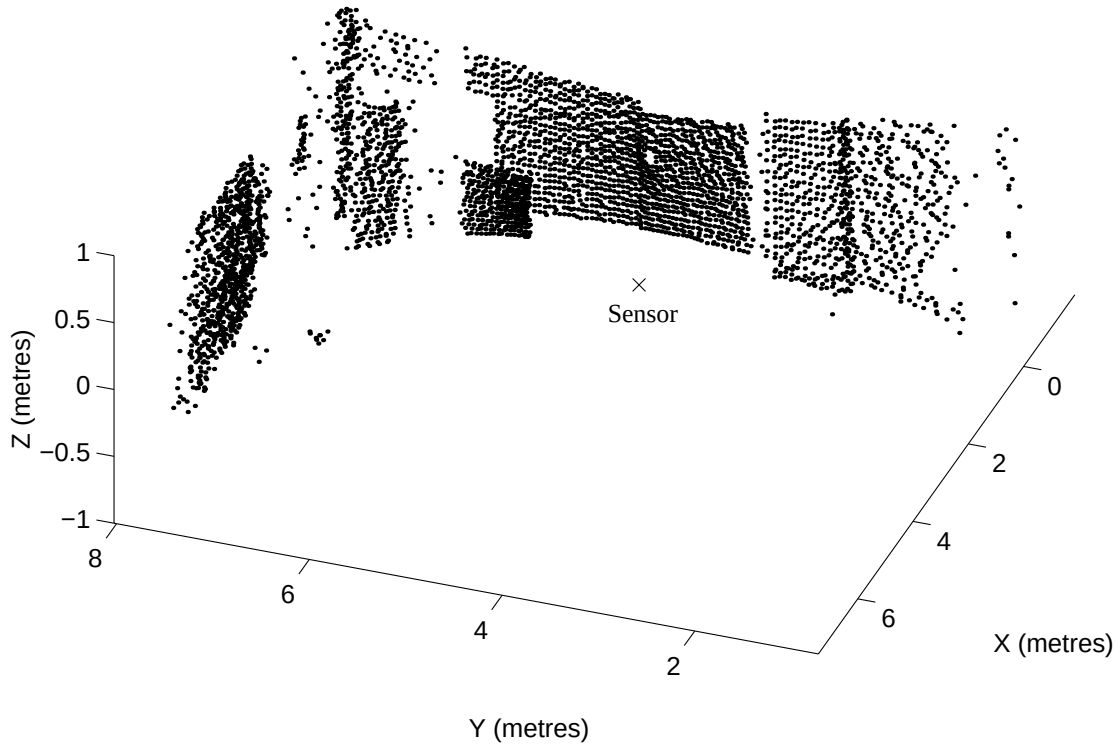

Figure 19: *A section of the 3D scan data after the robot has moved 80 cm in the negative y direction from the 'basis' scan point.*

is shown in order to give an example of the quality of the 3D data which is obtainable from a lidar sensor. In order to show the matched edges, amongst all of this scan data, figure 20 shows the same scan as figure 19 but viewed along the x axis, showing also the matched edges. The matched edges (12 in all), matched to the edges detected in figure 18, are shown as circles and are just visible along the line $y = 5.5$ and between $z = +0.18$ and $z = -0.33$. The sensor position is labeled and marked with a cross ($\times$). In total 12 successful matches were made using equation 31 in the appendix. The mobile robot was then moved 1.5m in the y direction taking it 0.7m to the other side of the 'basis' position. A new 3D scan was recorded and once again, to give a perspective of the new position, this scan data is shown in figure 21. In this scan 13 successful matches were made and for clarity are shown in figure 22 along with the scan data viewed along the x axis (actually at a 10^o offset to the x axis in order to distinguish the sensor from the edges). Now that the sensor is nearer to the line producing the detected edges, the data appears more dense since the distance in the z direction between each point is reduced.

The actual results of our 3D scanning and matching can be seen in figure 23. This figure shows a plan view of the environment and the hand measured (triangles) and EKF estimated (points) positions of the mobile robot and the uncertainty ellipsi. The middle position just

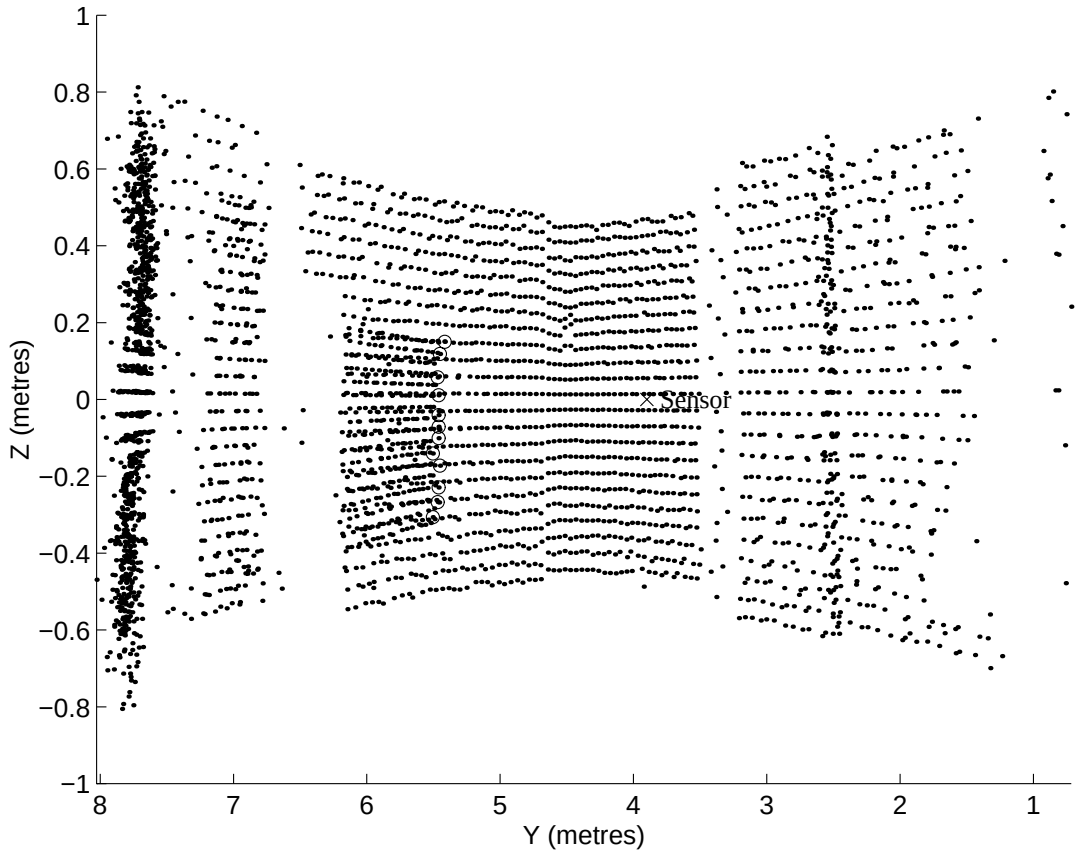

Figure 20: *The 3D scan data from figure 19 and matched edges (small circles) viewed along the x axis.*

shows the triangle (△) (hand measured position) since this is the 'basis' position used to detect edges for future matching purposes. Note that although we have focussed the edge detection algorithm into one region of the environment only, the positional uncertainty ellipsi in figure 23 are smaller than those recorded at the same positions when scanning in 2D only (see positions 12 in figures 16 and 17). Although in the 2D scans we had edges which resulted from a more diverse choice of environmental objects (with respect to the sensor's bearing angle), in the 3D case we recorded more edges from a particular environmental object. For example in figure 15 we see 4 matched edges at position 12, all from different objects within the environment. At the same position we matched 12 edges from one object when scanning in 3D (figure 20). Hence focusing the attention of the edge detection and matching process to a particular object or region within a 3D scan can be more beneficial for mobile robot localisation than attempting to detect all edges within a 2D scan and then match them.

It can be seen in figures 20 and 22 that as the sensor moves away from a feature from which edges are expected to be detected, the distance between the edges in the z direction increases. The 3D observation model presented in section 5.3 automatically accounts for this and mobile robot localisation is therefore still achieved. A problem which arises from this effect however is that the expected edge coordinates can leave the field of view of the sensor (if it moves to close to the object producing the edges) since their z coordinate will be too high or low due to the restricted scanning azimuth capability of the sensor. If the sensor moves to far from the object

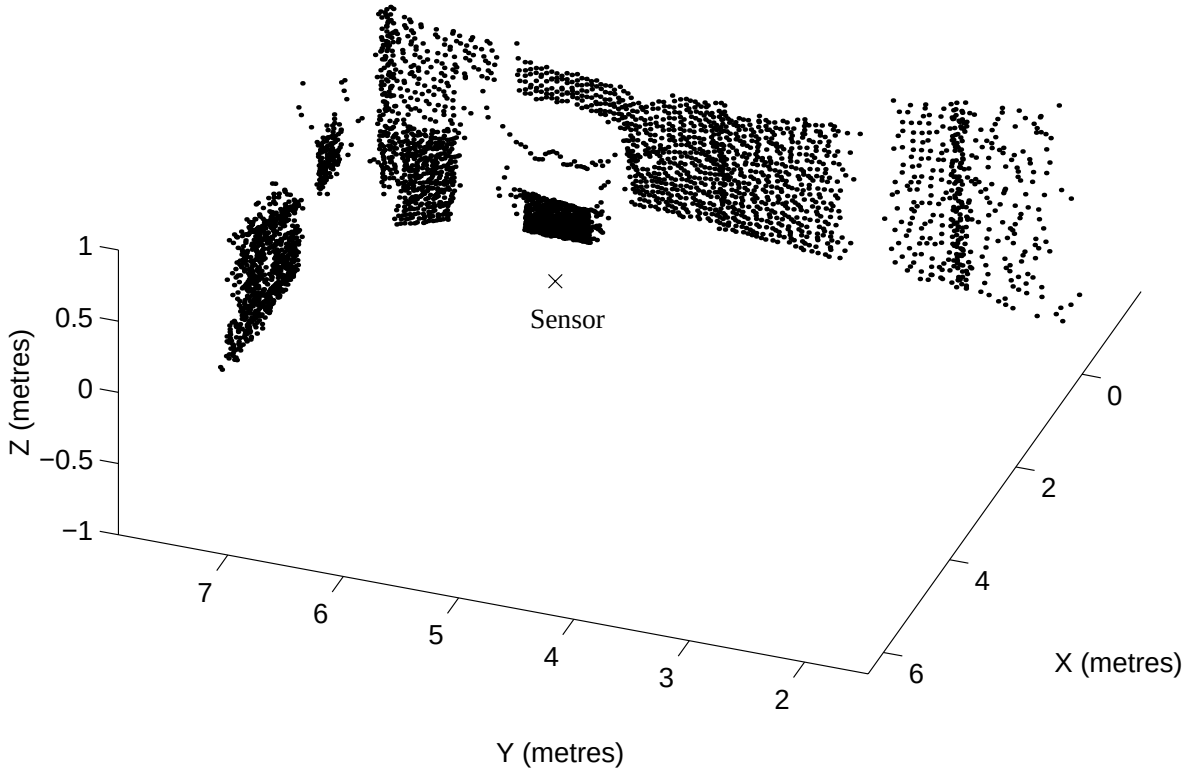

Figure 21: *A section of the 3D scan data recorded after the mobile robot moved 70cm to the other side of the 'basis' scan point.*

producing the edges, then very few of them will be detected due to the limited resolution in scanning azimuth (0.5^o in our case). Hence this 3D localisation technique is very powerful for accurately and reliably determining the position of a mobile robot, but can only be used for robot positions near to the 'basis' position. In practise we found that when the robot was more than 3m away from its 'basis' position, a new set of 'basis' edges had to be recorded to continue this form of 3D localisation.

8 Summary

We have presented here an approach to navigation which unifies the extraction of local information from a mobile robot's surroundings with the application of a kinematic model based localisation algorithm. In particular we have addressed the following issues:

1. We have analysed 2D and 3D optical range and intensity scans and analysed the physics behind the AMCW lidar sensing process in order to determine what information from our real sensor data could be used for robot navigation. We explained the relationship between the received signal strength recorded by these sensors and the variance in the

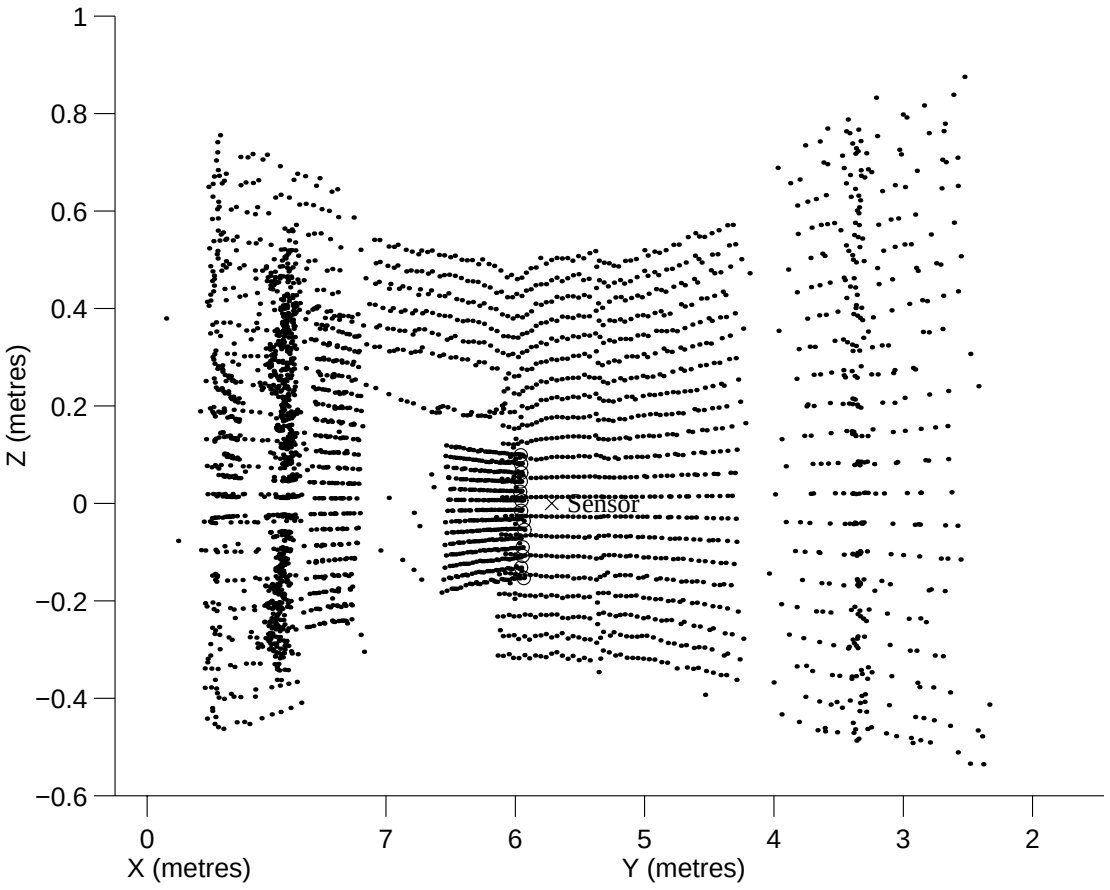

Figure 22: *The 3D scan data from figure 21 and matched edges (small circles) viewed along the x axis.*

range readings. We also demonstrated and gave reference to the solution of the problem of spurious data which results when the optical beam is split between objects of differing range and/or surface reflectivity.

2. We presented a novel 3D edge detector capable of finding, *on line* (ie: as the data arrives), the coordinates of edges within an indoor environment. The edge detector comprises a simple 2 state EKF and makes optimal predictions of edge coordinates (in the minimum mean squared error sense) based upon each range sample and its range variance. We have quantified the effects of various parameters within the edge detection algorithm.

3. We derived localisation algorithms which relate both 2D and 3D detected edges to the true position of a mobile robot. These algorithms also exploit the EKF in order to produce an optimal estimate of a mobile robot's position and a region of certainty surrounding this estimate, which is known to contain the robot with a predefined probability. The localisation techniques presented were divided into the following experiments:

   (a) *2D localisation with an a priori map.* We demonstrated a robust localisation algorithm which compared newly detected edges to an *a priori* map. This *a priori* map was defined simply as a list of edge coordinates which a human would expect the

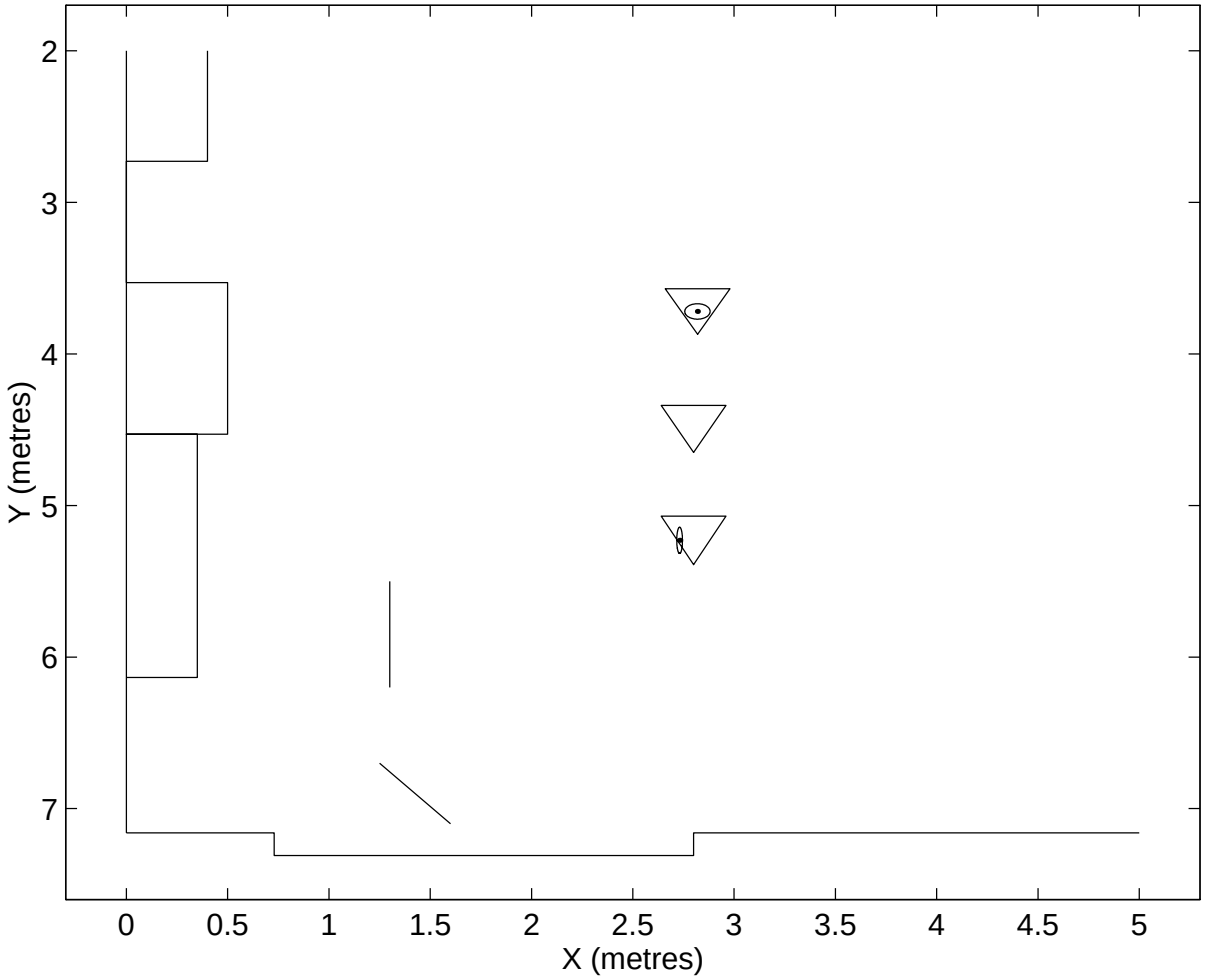

Figure 23: *A plan view of the model environment from figure 18, showing the robot basis position (middle triangle) and the hand measured positions (triangles), EKF estimated positions (points) and, centred at these points, the uncertainty ellipsi for the other two positions.*

edge detection algorithm to find. The EKF estimated positions of the mobile robot were much closer to the actual positions than those estimated by the mobile robot's odometric system.

(b) *2D localisation without an a priori map.* We showed the generality of the localisation algorithms presented here as it was possible to match newly detected edges to edges which had been found in previous scans. This is a dynamic process: as new edges are detected they are stored for future matching purposes and as old edges are no longer useful for matching they are eventually removed from memory. This method is slightly more sensitive to spurious data (resulting from a split optical beam) than the above method, but, in a limited context, also provides estimated robot positions which are much closer to the actual positions than those estimated by the mobile robot's odometric system. After many moves, precisely known features (i.e. some form of simple *a priori* map) are still required.

(c) *3D localisation.* Finally we demonstrated the localisation algorithms using 3D range data. We showed that 3D scanning can produce much more accurate positional estimates than 2D scanning. 3D localisation can be a very accurate and efficient technique if the edge detection algorithm is focussed to run in particular regions of the environment so that a limited number of new edge coordinates are recorded in each scan for future matching purposes.

Appendix - The Extended Kalman Filter (EKF)

We explain here the EKF algorithm for providing optimal state estimates in the least squares sense. We also explain here how our observations can be validated.

The EKF is used to produce an "optimum" estimation of the state $\hat{\mathbf{x}}(k+1 \mid k+1)$ by using as much "sensible" sensor data as possible[5]. This is done in the presence of sensor measurement noise. The algorithm or filter cycle is explained below, whilst the derivation of the equations used is in (Bar-Shalom and Fortmann 88).

1. *Filter Initialisation.* Initialise the *mean square error covariance matrix* $\mathbf{P}(j \mid j)$ and predicted state $\hat{\mathbf{x}}(j \mid j)$, for some value of j. (This is discussed in (Bozic 86)).

2. *Filter Prediction.* Determine the predicted state and error covariance matrix using the *Kalman Predictor* equations:

$$\hat{\mathbf{x}}(k+1 \mid k) = \mathbf{F}(\hat{\mathbf{x}}(k \mid k), \mathbf{u}(k)) \tag{19}$$

(Bar-Shalom and Fortmann 88) shows that in order to obtain the predicted error in the states, namely the error covariance matrix $\mathbf{P}(k+1 \mid k)$ - ie: the value of $\mathbf{P}$ at time $k+1$ given all observations up to and including time k, the non-linear function $\mathbf{F}$ has to be linearised about the prediction $\mathbf{x} = \hat{\mathbf{x}}(k \mid k)$. Hence we expand $\mathbf{F}$ as a Taylor series about the previous prediction $\hat{\mathbf{x}}(k \mid k)$. $\mathbf{F}(\mathbf{x}(k))$ is therefore replaced by the Jacobian:

$$\mathbf{A} = \nabla \mathbf{F} = \left[\begin{array}{cccc} \frac{\partial x_1(k+1)}{\partial x_1} & \frac{\partial x_1(k+1)}{\partial x_2} & . & . \\ \frac{\partial x_2(k+1)}{\partial x_1} & \frac{\partial x_2(k+1)}{\partial x_2} & . & . \\ . & & . & \\ . & & & . \\ . & & & . \end{array} \right]_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} \tag{20}$$

giving the linearised system model:

$$\delta\mathbf{x}(k+1) \approx \mathbf{A}\delta\mathbf{x}(k) \tag{21}$$

where the $\mathbf{A}$ matrix is to be re-evaluated at every new time step k.

The predicted error covariance matrix is, in general, given by:

$$\mathbf{P}(k+1 \mid k) = \mathbf{A}\mathbf{P}(k \mid k)\mathbf{A}^T + \mathbf{Q}(k) \tag{22}$$

3. *Observation.* Obtain the next observation. In the case of the edge detection algorithm, this is simply the next range reading $z(k+1)$. In the case of the localisation algorithm, this observation consists of n_o observed edges in the environment and is given by:

$$Z(k+1) = \{\mathbf{z}_j(k+1) \quad | \quad 1 <= j <= n_o\}. \tag{23}$$

4. *Measurement (observation) Prediction.* Use the predicted state $\hat{\mathbf{x}}(k+1 \mid k)$ to generate (a) predicted observation(s). For the edge detection case this is simply:

$$\hat{z}(k+1) = \mathbf{C}\hat{\mathbf{x}}(k+1 \mid k) \tag{24}$$

[5]where $\hat{\mathbf{x}}(k+1 \mid k+1)$ means the "predicted value of state $\mathbf{x}$ at time $k+1$ given all observations up to and including time $k+1$".

For the case of mobile robot localisation we need a predicted observation for each of the n_p targets:

$$\hat{\mathbf{z}}_i(k+1) = \mathbf{h}(\mathbf{p}_t, \hat{\mathbf{x}}(k+1 \mid k)), i = 1, 2, ..., n_p \tag{25}$$

The target locations $\mathbf{p}_t$ are in a map or can be extracted from previous scans.

5. *Innovation.* Determine the innovation, the difference between the observation and the predicted observation. For the edge detection case this is simply:

$$v(k+1) = z(k+1) - \mathbf{C}\hat{\mathbf{x}}(k+1 \mid k) \tag{26}$$

$v(k+1)$ can be used to validate range measurements before they are incorporated into the filtered estimates.

For the localisation case, for each prediction i and observation j we compute the innovation $\mathbf{v}_{ij}(k)$:

$$\mathbf{v}_{ij}(k+1) = \mathbf{z}_j(k+1) - \hat{\mathbf{z}}_i(k+1) \tag{27}$$

6. *Variance of the Innovation.* Find the variance associated with the innovation. For the edge detector:

$$s(k+1) = \sigma_r^2(k+1) + \mathbf{C}\mathbf{P}(k+1 \mid k)\mathbf{C}^T \tag{28}$$

where $\sigma_r^2(k+1)$ denotes the range variance at time $k+1$.

For the localisation algorithm:

$$\mathbf{S}_{ij}(k+1) = \nabla\mathbf{h}_i\mathbf{P}(k+1 \mid k)\nabla\mathbf{h}_i^T + \mathbf{R}_i(k+1) \tag{29}$$

where $\mathbf{R}_i(k+1)$ denotes the observation error covariance matrix at time $k+1$.

7. *Validation Gate.* Use the innovation and innovation variance to form a validation gate equation. For the edge detector:

$$D(k+1) = v^2(k+1)s^{-1}(k+1) \tag{30}$$

and for the localisation algorithm:

$$D(k+1) = \mathbf{v}_{ij}(k+1)\mathbf{S}_{ij}^{-1}(k+1)\mathbf{v}_{ij}^T \tag{31}$$

The term $D(k+1)$ is sometimes also referred to as the normalised innovation. In the case of the edge detector, this simply defines a line of constant probability in observation space, centred on the prediction. A threshold value for $D(k+1)$ is then necessary so that we can label observations which fall outside of this region or gate as possible edges.

In the localisation algorithm, this equation is used to test each observation $z_j(k+1)$ for membership in the validation gate for each predicted measurement. Because only range measurements are included in this validation gate, we use another test to determine if an observation is valid or not. We use the angular uncertainty around an edge. This angular uncertainty can be determined from the error covariance (from equation 22 of an edge (figure 24).

We assume that the range variance of an edge can be extended in all directions (a circular region) around the edge. Together with the current range to the edge, we can determine this angular uncertainty.

$$\Delta\gamma = \tan^{-1}\frac{a}{r} \tag{32}$$

where a is the range variance and r the distance to the target in meters.

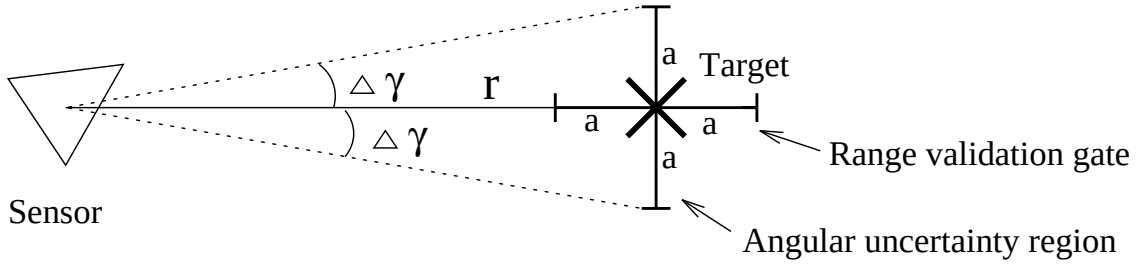

Figure 24: Determining angular uncertainty from the range variance.

8. *Define* $\mathbf{W}(k+1)$ - *The Kalman Gain.* In the edge detection case we define a new matrix as:

$$\mathbf{W}(k+1) = \mathbf{P}(k+1 \mid k)\mathbf{C}^T s^{-1}(k+1) \tag{33}$$

In the localisation case, after matching, a number of targets can be used to update the position of the mobile robot. To achieve this, the filter gain has to be calculated for every matched edge j:

$$\mathbf{W}_j = \mathbf{P}(k+1 \mid k)\nabla^T \mathbf{h}_j \mathbf{S}_j^{-1}(k+1) \tag{34}$$

9. *State vector update.* We use $\mathbf{W}(k+1)$ to update the state vector and the error covariance matrix. For the edge detector:

$$\hat{\mathbf{x}}(k+1 \mid k+1) = \hat{\mathbf{x}}(k+1 \mid k) + \mathbf{W}(k+1)v(k+1) \tag{35}$$

$$\mathbf{P}(k+1 \mid k+1) = \mathbf{P}(k+1 \mid k) - \mathbf{W}(k+1)s(k+1)\mathbf{W}^T(k+1) \tag{36}$$

For the localisation algorithm, the newly estimated vehicle location then follows from:

$$\hat{\mathbf{x}}(k+1 \mid k+1) = \hat{\mathbf{x}}(k+1 \mid k) + \sum_j \mathbf{W}_j(k+1)v_j(k+1) \tag{37}$$

where the summation is calculated over all matched edges.

The estimated covariance can be found from:

$$\mathbf{P}(k+1 \mid k+1) = \mathbf{P}(k+1 \mid k) - \sum_j \mathbf{W}_j(k+1)\mathbf{S}_j(k+1)\mathbf{W}_j^T(k+1) \tag{38}$$

10. *Recalculate* $\mathbf{A}$. We now recalculate $\mathbf{A}$ based on the new prediction $\hat{\mathbf{x}}(k+1 \mid k+1)$ and then return to step 2 to continue the cycle.

9 Acknowledgments

We would like to thank Prof. G. Schweitzer for supporting this work.

References

M. D. Adams and P. J. Probert, 1996. The interpretation of phase and intensity data from amcw light detection sensors for reliable ranging. *Int. J. Robotics Research*, 15(5):441–458.

M. D. Adams, 1992. *Optical Range Data Analysis for Stable Target Pursuit in Mobile Robotics.* PhD thesis, University of Oxford, United Kingdom.

M. D. Adams, 1996. *An Eye Safe Light Detection and Ranging Sensor for Scanning Indoor Environments.* Technical Report, Institute of Robotics, Swiss Federal Institute of Technology, Zurich.

Y. Bar-Shalom and T.E. Fortmann, 1988. *Tracking and Data Association.* Academic Press.

Besl P. J., 1988. *Surfaces in Range Image Understanding.* New York: Springer-Verlag.

J. Borenstein, H. R. Everett, and L. Feng, 1995. *Navigating Mobile Robots: Sensors and Techniques.* A. K. Peters, Ltd., Wellesley, MA.

S. M. Bozic, 1986. *Digital and Kalman Filtering.* Edward Arnold.

Canny J., 1986 (November). A computational approach to edge detection. In *IEEE Trans. Pattern Analysis Machine Intell.*, pages 679–697, vol. 8.

Connor F. R., 1982. *Noise.* London: Edward Arnold.

J.L. Crowley, 1985. Navigation for an Intelligent Mobile Robot. *IEEE J. Robotics and Automation*, 1:31–41.

Hebert M. and Krotkov E., 1991. 3-D measurements from imaging laser radars: How good are they? In *Int. Conf. Intelligent Robots and Systems*, pages 359–364.

Johnston A. R., 1973. *Infra-red Laser Rangefinder.* Technical Report, Jet Propulsion Laboratory, Pasadena, CA.

Erwin Kreysig, 1983. *Advanced Engineering Mathematics.* John Wiley & Sons, 5th edition.

Krotkov E., 1990. *Laser Rangefinder Calibration for a Walking Robot.* Technical Report, Robotics Institute, Carnegie Mellon University, Pittsburgh PA.

J. Leonard and H.F. Durrant-Whyte, 1990. Application of Multi-Target Tracking to Sonar Based Mobile Robot Navigation. In *International Conference on Decision and Control.*

John J. Leonard and Hugh F. Durrant-Whyte, 1992. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers.

M. J. Brownlow, 1993. *A Time of Flight Optical Range Sensor for Mobile Robot Navigation.* PhD thesis, University of Oxford, United Kingdom.

Miller G. L. and Wagner E. R., 1987. *An Optical Rangefinder for Autonomous Robot Cart Navigation.* Technical Report, A.T&T. Bell Laboratories, Princeton NJ.

Nitzan D., Brain A. E., and Duda R. O., 1977. The measurement and use of registered reflectance data in scene analysis. In *Proc. IEEE.*, pages 206–220, vol. 65.

R. Smith, M. Self, and P. Cheeseman, 1990. Estimating uncertain spatial relationships in robotics. *In Autonomous Robot Vehicles, editors: I. J. Cox and G. T. Wilfong*, 167–193.