SEGMENTS: A Layered, Dual-Kalman filter Algorithm for Indoor Feature Extraction*

Stergios I. Roumeliotis[1†] and George A. Bekey[1,2]

stergios|bekey@robotics.usc.edu

[1]Department of Electrical Engineering
[2]Department of Computer Science
Robotics Research Laboratories
University of Southern California
Los Angeles, CA 90089-0781

Abstract

A layered algorithm for extracting features in an indoor environment from planar range data is presented. At the lower (signal processing) level, the SEGMENTS algorithm exploits the speed and accuracy of two Extended Kalman filters working in tandem and processes sequentially the distance measurements within each scan. The first Kalman filter is responsible for initiating straight line segments and detecting clutter while the second estimates the parameters of each line segment (such as distance and orientation) and determines when this is interrupted. The dual filter combination is capable of detecting edges and straight line segments within the scene infront of the robot. At the higher (post-processing) level, the identified segments are combined to form more complex features such as extended walls, corners (concave and convex), doors and corridors. During the composition cycle, SEGMENTS makes use of (i) the parametric representation of the straight line segments and (ii) the prespecified topological models of the features this algorithm seeks. A list of the identified features along with their location with respect to the laser sensor is finally available to the user. The cluttered regions in the scene are also marked on a polar representation of the environment. The presented algorithm has been tested on a Pioneer 2 DX mobile robot equipped with a SICK, LMS 200 proximity laser scanner performing map-based localization. Low computational requirements, accuracy and robustness to uncertainty and noise characterize the performance of this new method for feature extraction.

1 Introduction

In order for a mobile robot to autonomously navigate, it has to be able to localize itself. The basic idea behind most of the current localization systems operating in a previously known indoor environment[1] is that a robot carries exteroceptive sensors to perceive the environment and match the sensed data with the expected data available in the form of a map of the area. This expectation-measurement cycle is used by the robot to update its position and correct the localization error due to odometry. In addition, the exteroceptive sensor information can be used to simultaneously localize the robot and construct a map of the environment along the robot's trajectory. In order to increase the efficiency and robustness of the localization and/or mapping process, the exteroceptive sensor data have to be transformed in a more compact form before attempting to compare them to the ones presented on an a previously known map, or store them in a currently constructed map. In either case, the precision and reliability of the task depends heavily on the form selected for the representation of the surroundings of the robot.

One popular category of approaches is to use a grid based representation of the environment and assign values to each individual cell according to the probability of this cell being occupied by an object or being part of the empty space ([6], [14] [22]). The main drawbacks of these approaches are (a) the major computational requirements, especially in large areas and (b) the dependence of precision on the size of the unit cell selected a priori by the designer. A different approach is to map the exteroceptive sensor data into a set of features commonly found in the environment of the robot [3], [11]. Unless the environment of the robot has been engineered to include artificial landmarks (beacons) [9], [2], the success of the second approach is conditioned on (a) the existence of accurate exteroceptive sensor(s) capable of sensing the environment in sufficient detail for discriminating between similar features, and (b) the availability of fast and reliable algorithms capable of extracting features from a large set of noisy and

*This work is partially supported by NASA-JPL (contract 959816) and DARPA (contracts F04701-97-C-0021 and DAAE07-98-C-L028).

†Contact author for correspondence.

[1]The Global Positioning System (GPS) cannot be used for global localization inside a building because in most cases the GPS signals are occluded and if not, the provided accuracy is not sufficient for tight navigation within a cluttered environment.

uncertain data.

Some of the early efforts in this direction have focused on extracting different structural features in an indoor environment based on the information provided by sonar range sensors. The multiple hypothesis tracking approach [4], [10], has been applied to encounter the data association problem, i.e. distinguish between reflections from corners, planes and cylinders. Sonar sensors suffer from (a) frequent specular reflections and (b) a significant spread of energy known as beamwidth. The large angular uncertainty and the secondary reflections make the problem of feature extraction quite complex [8]. Grid based maps and repetitive measurements (in order to view targets from different locations) are methods for dealing with the inherent limitations of a sonar [14], [16].

On the other end of the sensor phasma, vision (monocular, stereo, or trinocular) provides richer and more versatile information compared to most other sensors. Vision based methods for localization are described in the relative literature (e.g. [21], [7], [20]). Especially in an outdoor environment, well calibrated stereo vision is capable of collecting rich range data and inferring the motion of a robot from them [15], [12]. Applying vision to feature extraction leads to increased CPU usage due to the complexity of the algorithms required. The extensive processing for each frame collected is usually the limiting factor for many applications [17].

Since the structural features commonly found in an indoor environment are in general invariant with respect to height (e.g. walls, corners, doors), a planar representation with dense and accurate range measurements would be adequate for feature extraction. In this case, the processing requirements would be significantly less than those of a vision system due to the reduced dimensionality of the collected information. A laser range scanner[2] is capable of collecting such high quality range data and thus it is ideally suited for indoor environments. In addition, the laser beam involved in the calculation of the time of flight to an object, is not affected by the lighting conditions. Compared to an ultrasonic range sensor, the laser range sensor has the significant advantages of (a) very small number of specular reflections and (b) increased ranging and bearing accuracy. Since the angular uncertainty of the laser sensor is very small, a very fine description of the surroundings is available to the robot and thus structural features can be extracted directly from each scan without having to reposition the robot. Recently, the reduced cost of the laser scanner had made it an affordable device for many robotic systems.

2 Characteristics of the Laser Sensor

The LMS 200 laser sensor shown in Figure 1 has accuracy of approximately 15 mm (3 sigma value). The field of view is 180 degrees in front of the robot and up to 8 meters

[2]Also known as laser radar or lidar or ladar.

Figure 1: The Pioneer 2 DX mobile robot equipped with an LMS 200 laser scanner.

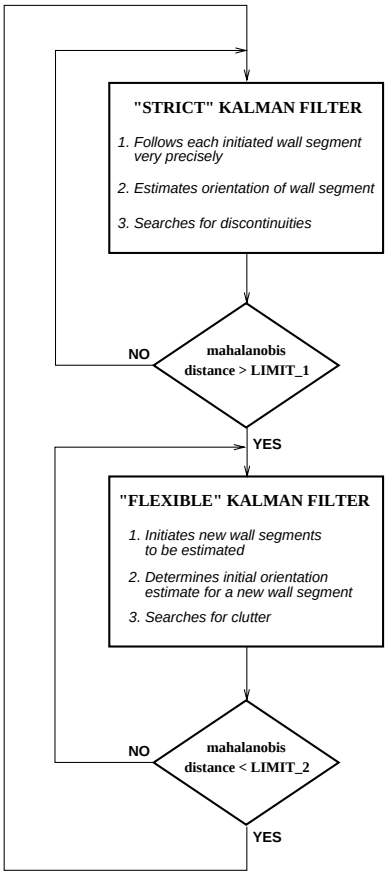

Figure 2: Block diagram of the two Extended Kalman filters sequentially processing the range measurements as these are recorded by the laser range sensor.

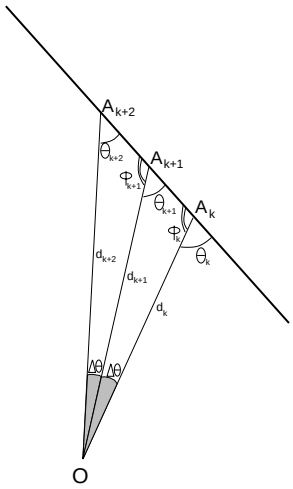

Figure 3: A planar laser scan from a flat wall.

of distance. The range samples are spaced every half a degree, all within the same plane. The sampling frequency was set at 5 Hz.

The surface of an object in the environment is visible to a certain level of detail depending on (a) the distance of the object from the laser scanner and (b) the relative orientation of the surface of the object with respect to the incidence angle of the laser beam. Objects close to the robot are very densely represented while remote surfaces (up to 8 meters) return approximately $8\pi/360$ reflection points per meter for a 90 degree angle of incidence. When the laser beam meets the surface at some angle θ, the number of points is reduced to 1 point every L meters:

$$L = \frac{sin(\Delta\theta)}{sin(\theta + \Delta\theta)}\ d \tag{1}$$

where $\Delta\theta = \pi/360$ and d is the distance from the first incident point on the surface. If R is the distance from the surface, then d is given by:

$$d = \frac{R}{sin\theta} \tag{2}$$

As the angle of incidence θ goes to zero (the laser beam is almost parallel to the surface), the distance from the next incident point goes to infinity. Thus, the number of reflection points per meter is reduced drastically and this surface becomes invisible to the laser scanner.

In the following section we give a brief description of two other approaches to range data processing suitable for laser scanner measurements. In Section 4 we present in detail the proposed feature extraction algorithm. Section 5 contains the experimental evaluation of the SEGMENTS algorithm and finally conclusions are part of Section 6.

3 Previous Approaches

In [5], a least-squares line fit was applied to extract edges from ultrasonic sensor data. Thresholding of the squared error between each range value and the computed straight line segment is used to select the points that belong in the line and determine the two bounding edge points. Instead of fitting straight line segments after a full scan has been recorded we choose to continuously process the range data as they arrive, similar to [1].

In [1] a Kalman filter is employed to process the incoming range measurements and detect edges to be used for localization. The equation that describes the system model of the filter is the equation of a straight line derived using three consecutive range points: $d_{k+2} = \frac{d_k\ d_{k+1}}{2\ d_k\ cos(\Delta\theta) - d_{k+1}}$. The state vector $\mathbf{x}_k = [d_{k+1}\ d_k]'$ contains the range estimates for two consecutive points within the laser scan. At each step the range estimate is compared to the range measurement in order to decide if an edge has been detected. Two different threshold values are used within this algorithm; one for indicating possible edges and a second (lower) for initializing a new line segment. Only large differences between measured and predicted range data are considered for indicating the position of an edge. The main drawback is that should this algorithm be extended to identify straight line segments in-between the edges, the orientation of these segments is not part of the state vector. Further combination of such segments would produce unreliable results. The relative orientation of neighboring line pieces is essential for constructing more complex structural features such as doors.

4 Feature Extraction Algorithm

The proposed SEGMENTS algorithm can be divided in two layers: signal processing and post-processing. The algorithm focuses on extracting information about the straight line segments contained in each scan. Both the orientation and the distance from the robot are being estimated.

4.1 Layer 1: Signal Processing Module

The essential components of the signal processing layer are two Extended Kalman filters tuned for two different approaches to line segment estimation. The same geometric description of line segments is incorporated in both filters. Each filter is designed to process the 361 data points within a scan sequentially as each range value appears starting from the -90 degrees reading towards the +90 degrees reading. The main difference is in the values of the parameters that describe how "strict" each filter is.

Initial trials with a single Kalman filter revealed that there is a trade-off between the sensitivity of a filter and its flexibility, i.e. its ability to adapt to new wall segments of different orientation within the same scan. A very "strict" filter is very sensitive to the presence of changes in the direction, but it takes a long time (many readings) to readjust to the new direction of a different wall segment. When a less "strict" filter is used, it can easily update its estimates and follow segments of different orientation but it can also miss progressive changes in orientation,

door frames, or even discontinuities due to the difference in width of two consecutive corridor segments.

In order to combine the advantages of both filters we introduce a new architecture that requires each filter to process different areas of a scan. By monitoring the filter residuals (differences between measured and expected range values), the algorithm decides which filter is more appropriate (better tuned) for each part of the scan. A well tuned filter tends to have residuals resembling zero-mean white noise signals.

The flow of control portrayed in Figure 2 is described by the following two steps:

1. The "strict" filter follows a lengthy straight line segment, filters the noisy range measurements and produces optimal estimates for the orientation and location of a wall. When a *discontinuity* appears the control is passed to the "less strict" or "flexible filter" (step 2).

2. The "flexible" filter attempts to follow a new wall segment of different orientation and/or location than the previous one. It uses a small number of range data points (three to four) to generate and test a new wall segment hypothesis. Every hypothesis contains a parametric description of a straight line segment. If this is successful (small residual), the control is passed again to the "strict" filter (step 1) for further refinement. If not, a new attempt with new data will take place (step 2 is repeated).

The main difference between the parameters of the two EKFs is in the values of the system noise covariance Q corresponding to the allowed deviation from the previous estimate in range and orientation. A larger value is selected for the "flexible" filter resulting in increased ability to initiate and follow new straight line segments.

4.1.1 Straight line segment equations

We now derive the equations for the model required for both Extended Kalman filters; i.e. a description of the straight line segments in terms of ϕ the complement to the angle of incidence θ and the distance from the point of incidence d.

Consider the straight line segment that is recorded when the 2D laser sensor scans a planar wall shown in Figure 3. The laser beam originating from point O intersects the wall segment at the consecutive points A_k, A_{k+1} and A_{k+2}. The distances from these points are d_k, d_{k+1} and d_{k+2}. The angles of incidence at these points are θ_k, θ_{k+1} and θ_{k+2}. Applying the law of sinusoids in the triangle $A_k O A_{k+1}$ we have:

$$\frac{d_k}{sin(\theta_{k+1})} = \frac{d_{k+1}}{sin(\phi_k)}$$

But $sin(\theta_{k+1}) = sin(\pi - \theta_{k+1}) = sin(\phi_{k+1})$ and thus:

$$d_{k+1} = \frac{sin(\phi_k)}{sin(\phi_{k+1})}\, d_k \tag{3}$$

The angle ϕ_{k+1} is an outside angle for the triangle $A_k O A_{k+1}$ and thus

$$\phi_{k+1} = \phi_k + \Delta\theta \tag{4}$$

where $\Delta\theta = \pi/360$ is the angle between two consecutive beams of the laser scanner. Equation (3) can now be written as

$$d_{k+1} = \frac{sin(\phi_k)}{sin(\phi_k + \Delta\theta)}\, d_k \tag{5}$$

4.1.2 Kalman filter equations

The state vector to be estimated is $\hat{\mathbf{x}}_k = [\hat{d}_k \;\; \hat{\phi}_k]$. The measurement vector is $\mathbf{z}_k = d_{meas}(k)$. The straight line model given by Equations (5) and (4) is non-linear, hence we employ the Extended form of the Kalman Filter (EKF). The EKF uses the system model and the measurement model. The system model describes where on the line the next data point $\mathbf{x}_{k+1} = \begin{bmatrix} d_{k+1} & \phi_{k+1} \end{bmatrix}'$ will appear in the presence of noise $\mathbf{v}_k$:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k, \tag{6}$$

where $\mathbf{f}$ is the non-linear transition function and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{Q}(k))$. The noise is assumed to be white, zero-mean Gaussian with variance $\mathbf{Q}(k)$.[3] The measurement model relates the sensor observations $\mathbf{z}(k) = \begin{bmatrix} d_{meas}(k) \end{bmatrix}'$ to the state of the system and has the following form:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k, \tag{7}$$

where $\mathbf{h}$ is, in general, a non-linear function and $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{R}(k))$. We assume that the measurements are corrupted by additive, white, zero-mean Gaussian noise with variance $\mathbf{R}(k)$.

The estimation algorithm produces an estimate of the state $\hat{\mathbf{x}}_{k+1,k}$ of the system at step $k+1$ (laser beam angle $k \times \Delta\theta$) based on the previously updated estimate of the state $\hat{\mathbf{x}}_{k,k}$ and the observations $\mathbf{z}_{k+1}$. The basic steps of the algorithm are: 1) Prediction, 2) Measurement, and 3) Estimation update. We state the Kalman filter equations without derivation and we refer to [13] for proofs and details.

1. Prediction: We predict the new state $\hat{\mathbf{x}}_{k+1,k}$ of the system using the system model at time step k:

$$\hat{\mathbf{x}}_{k+1,k} = \mathbf{f}(\hat{\mathbf{x}}_{k,k}). \tag{8}$$

[3] Since the beamwidth is very small, the angular uncertainty is very small and thus the noise component for angle ϕ is also very small. On the other hand, a straight wall segment is not a perfect straight line, the uncertainty regarding the distance from the point of incidence is large and thus the corresponding noise component for the distance d is also large (compared to the angular noise component).

and the covariance $\mathbf{P}(k+1,k)$ associated with this prediction:

$$\mathbf{P}(k+1,k) = \mathbf{F}(k)\mathbf{P}(k,k)\mathbf{F}^T(k) + \mathbf{Q}(k), \quad (9)$$

where $\mathbf{F}(k)$ is the Jacobian of $\mathbf{f}$ obtained by linearizing about the updated state estimate $\hat{\mathbf{x}}_{k,k}$:

$$\mathbf{F}(k) = \nabla \mathbf{f}(\hat{\mathbf{x}}_{k,k}). \quad (10)$$

In our case using Equations (5) and (4) we have:

$$\mathbf{F}(k) = \begin{bmatrix} \frac{sin(\phi_k)}{sin(\phi_k+\Delta\theta)} & \frac{sin(\Delta\theta)}{sin^2(\phi_k+\Delta\theta)}\ d_k \\ 0 & 1 \end{bmatrix} \quad (11)$$

Finally we compute the predicted measurements using the predicted state estimate $\hat{\mathbf{x}}_{k+1,k}$:

$$\hat{\mathbf{z}}_{k+1} = \mathbf{h}(\hat{\mathbf{x}}_{k+1,k}). \quad (12)$$

2. Measurements: In this step we get the actual measurements $\mathbf{z}_{k+1}$ and we compare them to the predicted ones $\hat{\mathbf{z}}_{k+1}$. The difference between them is the measurement residual (or innovation):

$$\mathbf{r}_{k+1} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1}. \quad (13)$$

We now calculate the covariance of the residual:

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1,k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1), \quad (14)$$

where $\mathbf{H}(k+1)$ is the Jacobian of $\mathbf{h}$ obtained by linearizing about the state estimate $\hat{\mathbf{x}}_{k+1,k}$:

$$\mathbf{H}(k+1) = \nabla \mathbf{h}(\hat{\mathbf{x}}_{k+1,k}) \quad (15)$$

and which for our case is:

$$\mathbf{H}(k+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (16)$$

3. Estimation update: In the last step of the estimation algorithm we use the measurement residual $\mathbf{r}_{k+1}$ to correct the state prediction $\hat{\mathbf{x}}_{k+1,k}$ and thus compute the updated state estimate $\hat{\mathbf{x}}_{k+1,k+1}$. In order to do that we calculate the Kalman gain[4]:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1,k)\mathbf{H}^T(k+1)\mathbf{S}^{-1}(k+1), \quad (17)$$

and then we update the state prediction:

$$\hat{\mathbf{x}}_{k+1,k+1} = \hat{\mathbf{x}}_{k+1,k} + \mathbf{K}(k+1)\mathbf{r}_{k+1}. \quad (18)$$

Finally we compute the associated covariance:

$$\mathbf{P}(k+1,k+1) = \mathbf{P}(k+1,k) - \mathbf{K}(k+1)\mathbf{S}(k+1)\mathbf{K}^T(k+1). \quad (19)$$

This will be used as the new state covariance in the next iteration of the estimation algorithm.

[4]Note that the matrix S is scalar and therefore the computationally expensive, for the Kalman filter, matrix inversion in Equation (17) is simplified to a division by scalar.

4.1.3 Validation gate

The previous equations describe both the "strict" and the "flexible" Kalman filter. Choice of different values for the system noise matrix $\mathbf{Q}(k)$ determines how strict or flexible each filter is. As mentioned before (Figure 2), every time a laser scan is processed by the dual-filter combination, the control is switched back and forth between the two filters. To decide which filter is in charge of processing each portion of the scanned image, we use the mahalanobis distance:

$$mahalanobis(k+1) = \mathbf{r}_{k+1}^T \mathbf{S}^{-1} \mathbf{r}_{k+1} \quad (20)$$

This is the validation gate equation (chi-square test) that allows the SEGMENTS algorithm to choose if the measured distance to the next observed point is close enough to the predicted distance and thus decide if it belongs in the straight line segment or not. The filter residuals, the difference between predicted and measured distances, are here used to test the convergence of the straight line-tracking Kalman filter. When this distance is greater than the limit chosen for the "strict" filter, this filter is not valid anymore. That is, the straight line segment being tracked has finished and a *discontinuity* is present in the data from the laser scanner scene. At this point (in the data) the "flexible" filter kicks in and resumes control of the estimation process. A new line segment is initiated and followed and the mahalanobis distance is calculated using the residuals from the "flexible" filter. A validation gate equation similar to Equation (20) is used to calculate the new mahalanobis distance. While this distance is larger compared to the limit associated with the "flexible" filter[5], a new straight line segment cannot be established and these point are reported as *discontinuities*. When the mahalanobis distance becomes smaller than the limit, the control is passed back again to the "strict" filter for further refinement of the newly extracted straight line segment.

4.2 Layer 2: Post-processing Module

When all the range measurements within a scan have been processed, the set of wall segments within a laser image is passed to the next layer (post-processing module) along with their locations, orientation and corresponding confidence to each of these values. The quality of these estimates will affect directly the performance of this module.

During post-processing, the SEGMENTS algorithm goes through the set of identified wall segments (separated from each other by one or more discontinuity points) and attempts to combine them into more complex features. Since it is required that at least 4 points constitute a straight line segment and there is at least 1 discontinuity point between two consecutive lines, the theoretical limit for the

[5]Both limits are not constant and their values depend on the estimated range and angle of incidence of each point of consideration. Even small errors in the angle estimate cause large errors when a distant point is encountered. An empirical formula is used for adjusting the value of each limit.

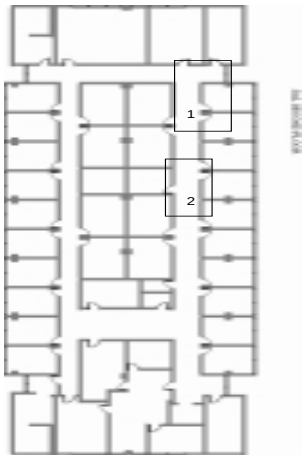

Figure 4: Sketch of the second floor of the Henry Salvatori Computer Center bldg. The experimental results correspond to the areas marked as 1 and 2.

maximum number of straight line segments within a scan is 72. In practice, this number is usually less than 10.

The reasoning for combining wall segments to structural features involves their geometric description and relations to other segments. A set of rules is applied to each subset of neighboring segments to determine if they constitute a corner or a closed door for example. Here, we report the rules applied for these two types of features only.

Conditions for the CORNER feature:

```
1 segment_length(i) > MIN_SEGMENT_LENGTH
2 segment_length(j) > MIN_SEGMENT_LENGTH
3 diff(segment_orientation(i),segment_orientation(j))
< MAX_DEVIATION_FROM_ORTHOGONAL
4 dist(end_segment(i),begin_segment(j))
< MAX_DEVIATION_FROM_CONSECUTIVE
```

Two segments constitute a corner if each segment's length is larger than a certain limit (Cond. 1, 2), they are orthogonal to each other (Cond. 3) and consecutive (Cond. 4).

Conditions for the DOOR feature:

```
1 segment_length(i) > MIN_SEGMENT_LENGTH
2 segment_length(j) > MIN_SEGMENT_LENGTH
3 segment_length(k) > MIN_SEGMENT_LENGTH
4 diff(segment_orientation(i),segment_orientation(j))
< MAX_DEVIATION_FROM_PARALLEL
5 diff(segment_orientation(j),segment_orientation(k))
< MAX_DEVIATION_FROM_PARALLEL
6 diff(segment_orientation(k),segment_orientation(i))
< MAX_DEVIATION_FROM_PARALLEL
7 dist(segment_position(i),segment_position(j))
> MIN_DEVIATION_FROM_COLINEAR
8 dist(segment_position(j),segment_position(k))
> MIN_DEVIATION_FROM_COLINEAR
9 dist(end_segment(i),begin_segment(j))
< MAX_DEVIATION_FROM_CONSECUTIVE
10 dist(end_segment(j),begin_segment(k))
< MAX_DEVIATION_FROM_CONSECUTIVE
```

Three segments can be combined to form a (closed) door if each segment's length is larger than a certain limit (Cond. 1, 2, 3), they are parallel (Cond 4, 5, 6), but not colinear (Cond. 7, 8), and they are consecutive (Cond. 9, 10). These conditions are part of $if < conditions > then < feature >$ rules employed to decide if there are features amongst the straight wall segments previously determined by the signal processing module.

5 Experimental Results

The SEGMENTS algorithm has been implemented on a mobile robot, shown in Figure 1, that performs map-based localization [18], [19]. Experimental data were collected in the corridors of an office environment shown in Figure 4. The laser scanner was placed on top of the robot at approximately 30 cm above the ground. At that level, clutter, such as trash cans and boxes populating the corridors, is visible to the robot. During the experiments people were walking around making the feature detection task even more challenging. The robot was placed in *area 1* (Figure 4) and commanded to move following the wall on its right side until it returns to the same point.

Since non existing features (false positives) can adversely affect the operation of the localization and/or mapping module, the SEGMENTS algorithm is tuned so as to minimize the number of false positives. During each trip, the robot encountered numerous doors and corners. Each of them was detected at least in one laser scan[6] while the number of false positives did not exceed one per trip.

Two representative examples of the performance of the SEGMENTS algorithm are shown in Figure 5. The data in the left and right column were collected in the areas 1 and 2, respectively, of Figure 4. The first row figures present the data points from the laser scanner after they have been prefiltered for specularities. The second row figures portray the discontinuities detected by the signal processing module. The third row figures contain the detected wall segments, doors and corners.

At this point we should mention that the robot was in motion while the laser sensor was scanning the environment for possible features. The 361 distances within one scan were not collected at the same time. There was a (up to 70 milliseconds) latency for collecting a full scan. This delay introduces additional errors due to the distortion of the range image of the environment.

[6] Reliable detection occurs within 3 to 4 meters from the laser scanner.

Compared to other line segment extraction algorithms, the implemented dual filter algorithm combines high sensitivity and high adaptability with very low computation requirements. In contrast to other estimation techniques that require iterative processing of the same laser scan, the SEGMENTS algorithm filters the range data sequentially as they appear and determines the parameters of the existing wall segments within a single pass. Post-processing of the results at a higher level is applied in order to search for features (composed of simple straight line segments) such as doors and corners.

6 Conclusions

In this paper a new algorithm for indoors feature extraction was presented. The SEGMENTS algorithm, at the lower level, employs two signal processing units (EKFs) tuned to perform two different types of estimation. The first ("flexible") filter is responsible for detecting straight line segments while the second one ("strict") is designed to estimate their parameters. At the higher level, rules are used by the post-processing module to combine these straight line segments, to structural features such as doors and corners.

This algorithm can provide a variety of different features to the mapping and localization procedures thus reducing the time required for a robot with unknown initial position to succesfully globally localize itself ([18]). The accuracy and robustness of the proposed method was demonstrated in a real world scenario while meeting real-time requirements.

References

[1] M.D. Adams and A. Kerstens. Tracking naturally occurring indoor features in 2-d and 3-d with lidar range/amplitude data. *International Journal of Robotics Research*, 17(9):907–923, Sept. 1998.

[2] E. T. Baumgartner and S. B. Skaar. An autonomous vision-based mobile robot. *IEEE Transactions on Automatic Control*, 39(3):493–501, March 1994.

[3] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, April 1997.

[4] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, April 1994.

[5] J. L. Crowley. Navigation for an intelligent mobile robot. *Journal of Robotics and Automation*, RA-1(1):31–41, March 1985.

[6] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, June 1989.

[7] E. Krotkov. Mobile robot localization using a single image. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, volume 2, pages 978–983, Washington, DC, 14-19 May 1989.

[8] R. Kuc and Y.-D. Di. Intelligent sensor approach to differentiating sonar reflections from corners and planes. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, pages 329–333, Amsterdam, 8-11 Dec. 1987.

[9] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[10] D. Maksarov and H. Durrant-Whyte. Mobile vehicle navigation in unknown environments: a multiple hypothesis approach. *IEE Proceedings-Control Theory and Applications*, 142(4):385–400, July 1995.

[11] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312, 1992.

[12] L. Matthies, B. Chen, and J. Petrescu. Stereo vision, residual image processing and mars rover localization. In *Proceedings of the 1997 IEEE International Conference on Image Processing*, volume 3, pages 248–251, Santa Barbara, CA, 26-29 Oct. 1997.

[13] P. S. Maybeck. *Stochastic Models, Estimation and Control*, volume 141-1 of *Mathematics in Science and Engineering*, chapter 6. Academic Press, 1979.

[14] H.P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pages 116–121, St. Louis, MO, 1985.

[15] C.F. Olson and L.H. Matthies. Maximum likelihood rover localization by matching range maps. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 272–277, Leuven, Belgium, 16-20 May 1998.

[16] D. Pagac, E. M. Nebot, and H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4):623–629, August 1998.

[17] J.A. Perez, J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardos. Continuous mobile robot localization: vision vs. laser. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, volume 4, pages 2917–2923, Detroit, MI, 10-15 May 1999.

[18] S.I. Roumeliotis. Reliable mobile robot localization. Technical Report IRIS-99-374, University of Southern California, April 1999. (http://iris.usc.edu/ irislib/).

[19] S.I. Roumeliotis and G.A. Bekey. Bayesian estimation and kalman filtering: A unified framework for mobile robot localization. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 2985–2992, San Fransisco, CA, April 24-28 2000.

[20] R. Sim and G. Dudek. Learning visual landmarks for pose estimation. In *Proceedings of the 1999 IEEE International Conference in Robotics and Automation*, volume 3, pages 1972–1978, Detroit, MI, 10-15 May 1999.

[21] K. Sugihara. Some location problems for robot navigation using a single camera. *Computer Vision, Graphics, and Image Processing*, 42(1):112–129, April 1988.

[22] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, Feb. 1998.

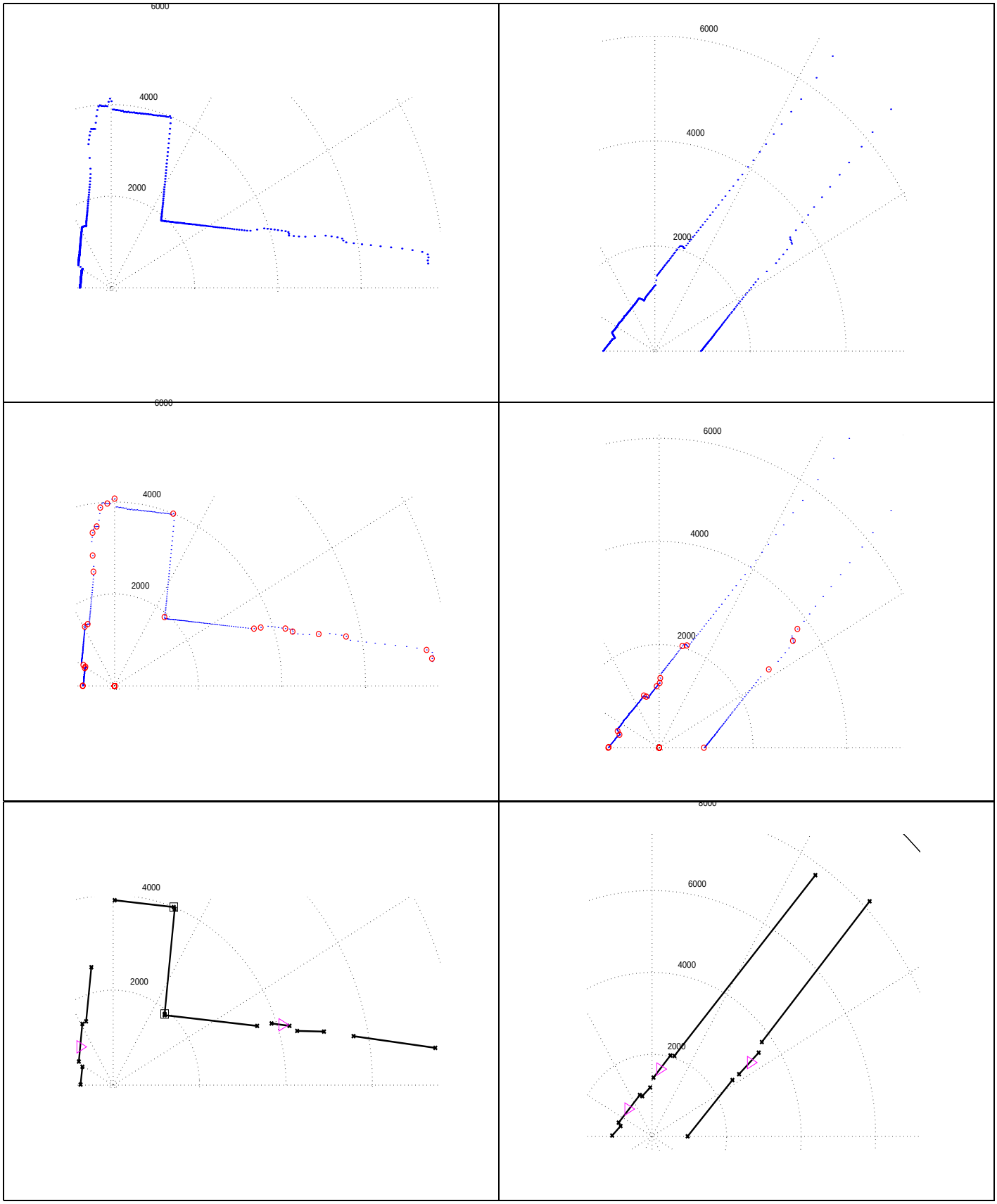

Figure 5: First row: Prefiltered laser scan range data. Second row: Discontinuities (edges and clutter). Third row: Straight line segments, doors (triangles), and corners (squares).