Technical report from Automatic Control at Linköpings universitet

Estimation of the Free Space in Front of a Moving Vehicle

Christian Lundquist, Thomas B. Schön, Umut Orguner

Division of Automatic Control

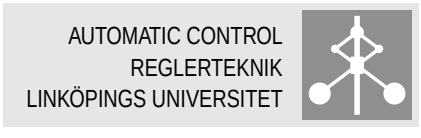
E-mail: lundquist@isy.liu.se, schon@isy.liu.se, umut@isy.liu.se

6th April 2009

Report no.: LiTH-ISY-R-2892
Accepted for publication in Proceedings of SAE 2009 World Congress, Detroit, MI, USA, April 20-23, 2009 and
Proceedings of IEEE Intelligent Vehicles Symposium, Xi'an, China, June 3-5, 2009

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se



Abstract

There are more and more systems emerging making use of measurements from a forward looking radar and a forward looking camera. It is by now well known how to exploit this data in order to compute estimates of the road geometry, tracking leading vehicles, etc. However, there is valuable information present in the radar concerning stationary targets, that is typically not used. The present work shows how radar measurements of stationary targets can be used to provide a reliable estimate of the drivable space in front of a moving vehicle.

In the present paper three conceptually different methods to estimate stationary objects or road borders are presented and compared. The first method considered is occupancy grid mapping, which discretizes the map surrounding the ego vehicle and the probability of occupancy is estimated for each grid cell. The second method applies a constrained quadratic program in order to estimate the road borders. The problem is stated as a constrained curve fitting problem. The third method associates the radar measurements to extended stationary objects and tracks them as extended targets.

The required sensors, besides the standard proprioceptive sensors of a modern car, are a forward looking long range radar and a forward looking camera. Hence, there is no need to introduce any new sensors, it is just a matter of making better use of the sensor information that is already present in a modern car. The approach has been evaluated and tested on real data from highways and rural roads in Sweden and the results are very promising.

Keywords: SLAM, Occupancy Grid Map, Guard Rail Detection, Extended Object Tracking

Estimation of the Free Space in Front of a Moving Vehicle

Christian Lundquist, Thomas B. Schön, Umut Orguner

2008-04-06

Abstract

There are more and more systems emerging making use of measurements from a forward looking radar and a forward looking camera. It is by now well known how to exploit this data in order to compute estimates of the road geometry, tracking leading vehicles, etc. However, there is valuable information present in the radar concerning stationary targets, that is typically not used. The present work shows how radar measurements of stationary targets can be used to provide a reliable estimate of the drivable space in front of a moving vehicle.

In the present paper three conceptually different methods to estimate stationary objects or road borders are presented and compared. The first method considered is occupancy grid mapping, which discretizes the map surrounding the ego vehicle and the probability of occupancy is estimated for each grid cell. The second method applies a constrained quadratic program in order to estimate the road borders. The problem is stated as a constrained curve fitting problem. The third method associates the radar measurements to extended stationary objects and tracks them as extended targets.

The required sensors, besides the standard proprioceptive sensors of a modern car, are a forward looking long range radar and a forward looking camera. Hence, there is no need to introduce any new sensors, it is just a matter of making better use of the sensor information that is already present in a modern car. The approach has been evaluated and tested on real data from highways and rural roads in Sweden and the results are very promising.

Contents

1 Introduction

For a collision avoidance system it is imperative to have a reliable *map* of the environment surrounding the ego vehicle. This map, consisting of both stationary and moving objects, has to be built in real time using measurements from the sensors present in the ego vehicle. This is currently a very active research topic within the automotive industry and in many other areas as well. Great progress has been made, but much remains to be done. Current state-of-the-art when it comes to the problem of building maps for autonomous vehicles can be found in the recent special issues [5, 6, 7] on the 2007 DARPA Urban Challenge. In these contributions measurements from expensive and highly accurate sensors are used, while we in the present report utilize measurements from off-the-shelf automotive sensors already present in modern cars.

Obviously, these stationary radar measurements are not enough to fully explain the road borders. However, as we will see, there is surprisingly much information present in these measurements.

The present solutions makes use of an already existing sensor fusion framework [25], which among other things provide a good road geometry estimate. This framework improves the raw vision estimate of the road geometry by fusing it with radar measurements of the leading vehicles. The idea is that the motion of the leading vehicles reveals information about the road geometry [45, 12, 13]. Hence, if the leading vehicles can be accurately tracked, their motion can be used to improve the road geometry estimates. Furthermore, we used a solid dynamic model of the ego vehicle allowing us to further refine the estimates by incorporating several additional proprioceptive sensor measurements readily available on the CAN bus. The resulting, rather simple, yet useful *map* of the environment surrounding the ego vehicle consists in

- Road geometry, typically parameterized using road curvature and curvature rate.
- Position and velocity of the leading vehicles.
- Ego vehicle position, orientation and velocity.

This information can be and has indeed been used to design simpler collision avoidance systems. However, in order to devise more advanced systems, more information about the environment surrounding the ego vehicle is needed. The purpose of this work is to exploit information already delivered by the radar sensor in order to compute a more complete map. Hence, there is no need to introduce any new sensors, it is just a matter of *making better use of the sensor information that is already present* in a modern premium car. To be more precise, it is the radar echoes from stationary objects that are used to estimate the road borders, which determines the free space in front of the ego vehicle. The radar measurements used originate from for instance, guard rails and concrete walls. Obviously these stationary radar measurements are not enough to fully explain the road borders. However, as we will see, there is surprisingly much information present in these measurements.

In the present report three conceptually different methods to estimate stationary objects or road borders are presented and compared. The first method described is occupancy grid mapping, which discretizes the map surrounding

the ego vehicle and a probability of occupancy is estimated for each grid cell. The second method applies a quadratic program in order to estimate the road borders. The problem is stated as a constrained curve fitting problem and is originally described in [27]. The third method associates the radar measurements to extended stationary objects and tracks them as extended targets.

The approach has been evaluated on real data from highways and rural roads in Sweden. The test vehicle is a Volvo S80 equipped with a forward looking 77 GHz mechanically scanned FMCW radar and a forward looking camera.

1.1 Related Work

The problem of road border estimation has been investigated in the literature during the last decade. The approaches presented mainly differ in their models for the road borders and the different types of sensors used in the estimation. The work presented here is clearly related to lane tracking, which by now is a very well-studied problem, see e.g. [30] for a recent survey using cameras. In fact the required sensor fusion framework [25] makes use of the estimates from a visual lane tracker. The recent book [8] contains a lot of interesting information about detecting and tracking lanes using cameras. Lane tracking has also been tackled using radar sensors, see e.g., [18,24,31,28] and laser sensors, see e.g. [44], which utilizes a linear model represented by its midpoint and orientation (one for each side of the road) for tracking road-curbs. Later, [23] enhanced the results of [44] with the addition of image sensors. A similar extended Kalman filtering based solution is given in [10] in a circular road border modeling framework. Recently, the particle filters (referred to as condensation in image and video processing) have been applied to the road border estimation problem in [42] with an hyperbolic road model.

Using laser scanners there have been several approaches making use of reflections from the road boundary, such as crash barriers and reflection posts, to compute information about the free space, see e.g. [21,20,36]. The third order approximation of the two sided (left and right) "clothoid model" has been used in connection with Kalman filters in [32] for radar measurements. The linear measurement model in Section 3 is similar to the one given in [32]. The two sided road border model that is used in almost all applications (and in [32]) is, however, quite conservative and can result in biased estimates especially in junctions and road separations in the highways. Our approach gives much more flexibility to road borders by allowing them to be appear wherever suitable.

Furthermore, the use of a side looking radar to measure the lateral distance to a sidewall is described in various papers, see e.g., [29,37,11]. The intended application in these papers [29,37,11] were automatic lateral control. Here, we have no specific application in mind, we just try to obtain the best possible map based on the available sensor information. This map can then be used by any control system.

In [43] an algorithm for free space estimation, capable of handling non-planar road's, using a stereo camera system is presented. It is similar to the present work, which make use of a parametric model of the road ahead. An interesting avenue for future work is to combine the idea presented in this paper with the ideas of [43] within a sensor fusion framework.

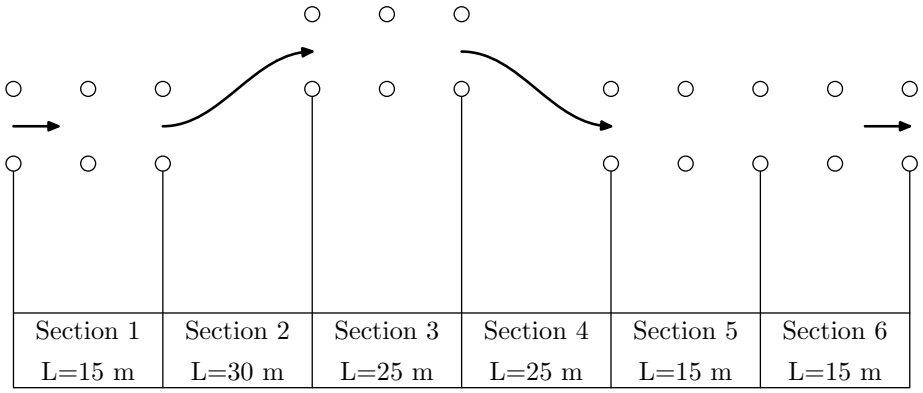


Figure 1: ISO 3888 double lane change maneuver [17].

1.2 Background and Notation

An important question is how the information of the free space should be represented and for which distances ahead of the vehicle that it is needed. We will start by addressing the latter through an example, the standard double lane change manoeuvre according to ISO 3888 [17]. In this maneuver a vehicle has to overtake an obstacle and come back to its original lane as shown in Figure 1. Assume that the ego vehicle is entering section 1 at a velocity of 100 km/h and that there is an obstacle straight ahead in section 3. The free space, i.e. the distance to the left and right road borders has to be known in order to autonomously overtake the obstacle as shown in the figure. This means that an automatic collision avoidance system needs to have information about the free space at least three sections ahead in order to make a decision on where to steer the vehicle. From this simple, yet informative, calculation we conclude that the road must be well estimated for at least 60 m ahead when driving at approximately 100 km/h.

In [25] we provide a sensor fusion framework for sequentially estimating the parameters l, δ_r, c_0 in the following model of the road's white lane markings,

$$y^E = l + \delta_r x^E + \frac{c_0}{2}(x^E)^2, \tag{1}$$

where x^E and y^E are expressed in the ego vehicle's coordinate frame E. The angle between the longitudinal axis of the vehicle and the road lane is δ_r, see Figure 2. It is assumed that this angle is small and hence the approximation $\sin\delta_r \approx \delta_r$ is used. The curvature parameter is denoted by c_0 and the offset between the ego vehicle and the white lane is denoted by l.

In this paper we will use the planar coordinate rotation matrix

$$R^{WE} = \begin{bmatrix} \cos\psi_{EW} & -\sin\psi_{EW} \\ \sin\psi_{EW} & \cos\psi_{EW} \end{bmatrix} \tag{2}$$

to transform a vector, represented in the ego vehicle's coordinate frame E, into a vector, represented in the world reference coordinate frame W, where ψ_{EW} is the angle of rotation from W to E. We will refer to this angle as the yaw angle of the vehicle, and in order to simplify the notation we will use $\psi \triangleq \psi_{EW}$.

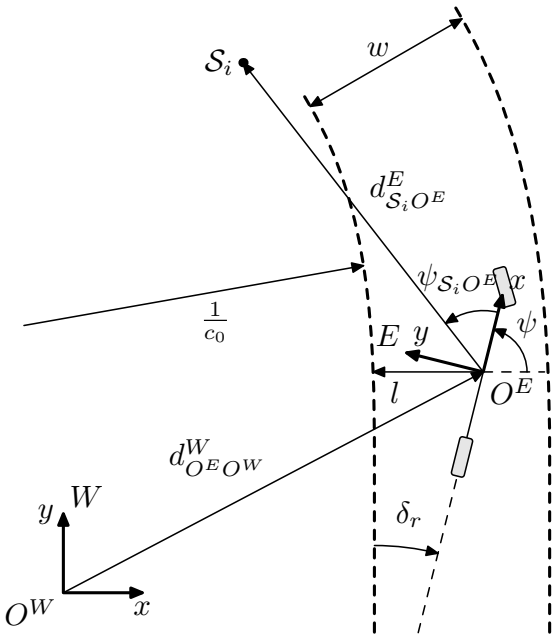


Figure 2: The ego vehicle's coordinate frame E has its origin O^E situated in the vehicle's center of gravity. A stationary object $\mathcal{S}_i$ is observed at a distance $||d^E_{\mathcal{S}_i O^E}||_2$ and an angle $\psi_{\mathcal{S}_i O^E}$ with respect to the radar. The angle between the vehicle's longitudinal axis and the road lane is δ_r, the distance to the left lane marking is l and the road curvature is c_0.

The point O^W is the origin of W and O^E is the origin of E situated in the vehicles center of gravity. The geometric displacement vector $r^W_{O^E O^W}$ is the straight line from O^W to O^E represented with respect to the frame W. The angles and distances are shown in Figure 2. Hence, a point P^E represented in the ego vehicle coordinate system E is transformed to be represented in the world coordinate system W using

$$P^W = R^{WE} P^E + r^W_{O^E O^W} \tag{3}$$

An observation m will be referred to as a stationary object in the point $\mathcal{S}_m$. The radar in the ego vehicle measures the azimuth angle $\psi_{\mathcal{S}E}$ and the range $d = ||d^E_{\mathcal{S}_m O^E}||_2$. These are transformed into Cartesian coordinates

$$\mathcal{S}^E_m = \begin{bmatrix} x^E_{\mathcal{S}_m O^E} & y^E_{\mathcal{S}_m O^E} \end{bmatrix}^T. \tag{4}$$

The nomenclature used in this work is summarized in Table 1. We will use quasi MATLAB notation describing the algorithms. This means that we use standard MATLAB notation and functions, but we keep the variable names as far as possible.

1.3 Report Outline

In the subsequent sections, we will analyze three different approaches to estimate the area in front of the ego vehicle.

Table 1: Nomenclature used in this work. The coordinate frame, in which a variable or constant is represented, is indicated with superscript. Models are indicated using calligraphic style. Variables or constants of a certain model are denoted by the model abbreviation subscript. The state descriptions are bold and sets are represented using normal font.

Abbr.	**Explenation**
c_0	road curvature
d	range
$d^W_{O^E O^W}$	line from O^E to O^W, in the W-frame
δ_r	angle between the vehicle's long. axis and the lane
E	ego vehicle coordinate frame
L	line coordinate frame
$\mathcal{L}$	modeled line
l	offset between the vehicle sensor and the left lane
N	number of elements
O^E	origin of E, at the vehicle's center of gravity
O^L	origin of L
O^W	origin of W
$\mathbf{P}$	state covariance
$\mathcal{P}$	modeled point
ψ	the ego vehicles yaw angle
$\psi_{\mathcal{S}_i E}$	bearing angle between the ego vehicle and an object i
Q	process noise covariance
R	rotation matrix
R	measurement noise covariance
r	offset between the vehicle sensor and the right lane
S	innovation covariance
$\mathrm{S_l}$	ordered set of stationary objects, left side of road
$\mathrm{S_r}$	ordered set of stationary objects, right side of road
$\mathcal{S}$	stationary object
$\mathcal{S}_m$	coordinates of a measured stationary object
θ	road border parameters
W	world coordinate frame
w	road width
$\mathbf{x}_{\mathcal{P}_i}$	state of point i
$\mathbf{x}_{\mathcal{L}_i}$	state of line i
$x^W_{O^E O^W}$	x-coordinate of a line from O^E to O^W, in W-frame
$y^W_{O^E O^W}$	y-coordinate of a line from O^E to O^W, in W-frame
$\mathbf{y}$	measurement

- Occupancy grid mapping in Section 2 discretizes a map into a number of cells with an associated probability of occupancy. The map is represented by a matrix, with each element corresponding to a map cell.
- Curve fitting using a quadratic program in Section 3 concerns the problem of estimating a left and right road boarder line by using a simple curve fitting algorithm with constraints. The border lines are represented by two polynomials.
- Extended target is Section 4 associates measurements to extended objects, e.g., guard rails are very narrow but long objects and are tracked as lines. These objects are represented as state variables in a standard nonlinear filtering framework.

The three estimators mentioned above are not only conceptually different, they also represents the result in completely different formats.

Finally in Section 5 the three methods are compared and their advantages and disadvantages are summarized.

2 Occupancy Grid Map

We would like to compute a map of the surrounding environment as few variables as possible. A map is defined over a continuous space and it can be discretized with e.g., a grid approximation. The size of the map can be reduced to a certain area surrounding the ego vehicle. In order to keep a constant map size while the vehicle is moving, some parts of the map are thrown away and new pars are initiated. Occupancy grid mapping (OGM) is one method for tackling the problem of generating consistent maps from noisy and uncertain data under the assumption that the ego vehicle pose is known. These maps are very popular in the robot community, especially for all sorts of autonomous vehicles equipped with laser scanners, indeed several of the DARPA vehicles [5, 6, 7] used OGM's. This is because they are easy to acquire, and they capture important information for navigation. The OGM was introduced by Elfes [9] and a solid treatment can be found in the recent textbook [38].

We start this section by providing a brief introduction to occupancy grid maps, according to [38]. Using this theory and a sensor with high resolution usually gives a nice looking bird eye's view map. However since we use a standard automotive radar, producing only a few range and bearing measurements at every time sample, some modifications are introduced as described in the following sections.

2.1 Background

The planar map $\mathbf{m}$ is defined in the world coordinate frame W and is represented by a matrix. The goal of any occupancy grid mapping algorithm is to calculate the filtering probability density function (PDF) of the map

$$p(\mathbf{m}|\mathbf{y}_{1:t}, \mathbf{x}_{E,1:t}), \tag{5}$$

where $\mathbf{m}$ denotes the map, $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$ denotes the set of all measurements up to time t, and $\mathbf{x}_{E,1:t}$ denotes the path of the ego vehicle defined

through the discrete sequence of all previous positions. An occupancy grid map is partitioned into finitely many grid cells

$$\mathbf{m} = \{m_i\}_{i=1}^{N_m}. \tag{6}$$

The probability of a cell being occupied $p(m_i)$ is specified by a number ranging from 1 for occupied to 0 for free. The notation $p(m_i)$ will be used to refer to the probability that a grid cell is occupied. A disadvantage with this design is that it does not enable us to to represent dependencies among neighboring cells.

The occupancy grid mapping was originally developed to primarily be used with measurements from a laser scanner. A laser is often mounted on a rotating shaft and generates a range measurement for every angular step of the mechanical shaft, i.e. a bearing angle. This means that the continuously rotating shaft produces many range and bearing measurements during every cycle. The OGM algorithms transform the polar coordinates of the measurements into Cartesian coordinates in a fixed world or map frame. After completing one mechanical measurement cycle the sensor provides the measurements for use.

The algorithm loops through all cells and increases the occupancy probability $p(m_i)$ if the cell was occupied according to the measurement $\mathbf{y}_t$. Otherwise the occupancy value either remains unchanged or is decreased, depending on if the range to the cell is greater or less than the measured range. The latter implies that the laser beam did pass this cell without observing any obstacles. If the measured range is great or the cell size is small, it might be necessary to consider the angular spread of the laser beam and increase or decrease the occupancy probability of several cells with respect to the beam width.

The map is supposed not to be changed during sensing, problems of this kind where a state does not change over time are often referred to as binary Bayes filter. In our case the state can either be free $m_i = 0$ or occupied $m_i = 1$. A standard technique to avoid numerical instabilities for probabilities near to 0 and to avoid truncation problems close to 0 and 1 is to use the log odds representation of occupancy

$$\ell_{i,t} = \log \frac{p(m_i|\mathbf{y}_{1:t}, \mathbf{x}_{E,1:t})}{1 - p(m_i|\mathbf{y}_{1:t}, \mathbf{x}_{E,1:t})}, \tag{7}$$

where the odds of a state is defined as the ratio of the probability of this event $p(m_i|\mathbf{y}_{1:t}, \mathbf{x}_{E,1:t})$ divided by the probability of its complement $1-p(m_i|\mathbf{y}_{1:t}, \mathbf{x}_{E,1:t})$. The probabilities are easily recovered using

$$p(m_i|\mathbf{y}_{1:t}, \mathbf{x}_{E,1:t}) = 1 - \frac{1}{1 + \exp \ell_{i,t}}. \tag{8}$$

Note that the filter uses the inverse measurement model $p(\mathbf{m}|\mathbf{y}, \mathbf{x})$. Using Bayes rule it can be shown that the binary Bayes filter in log odds form is

$$\ell_{i,t} = \ell_{i,t-1} + \log \frac{p(m_i|\mathbf{y}_t, \mathbf{x}_{E,t})}{1 - p(m_i|\mathbf{y}_t, \mathbf{x}_{E,t})} - \log \frac{p(m_i)}{1 - p(m_i)}, \tag{9}$$

where $p(m_i)$ represents the prior probability. The log odds ratio of the prior before processing any measurements is defined as

$$\ell_0 = \log \frac{p(m_i)}{1 - p(m_i)} \tag{10}$$

Typically $p(m_i) = 0.5$ is assumed, since before having measurements we know nothing about the surrounding environment. This value yields $\ell_0 = 0$.

2.2 OGM with Radar Measurements

The radar system at hand[1] provides range and bearing measurement for observed targets in two vectors at every measurement cycle. The main difference to a laser is that there is not one range measurement for every angular position of the moving sensor. The number of observations depends on the environment. In general there are much fever observations in comparison with standard laser systems. There is also a limit on the number of objects transmitted by the radar equipment on the CAN-bus. Moving objects, which are distinguished by measurements of the Doppler shift, are prioritized and more likely to be transmitted than stationary objects. We also assume that the opening angle of the radar beam is small in comparison to the grid cell size. These facts leads us to change the OGM algorithm to loop through the measurements instead of the cells, in order to decrease the computational load. A radar's angular uncertainty is usually larger than its range uncertainty. When transforming the polar coordinates of the radar measurements into the Cartesian coordinates of the map, the uncertainties can either be transformed in the same manner or it can simply be assumed that the uncertainty increases with the range.

2.2.1 The Algorithm

The main steps in a general OGM algorithm are given in Algorithm 2.1. In the first step the size and orientation of the map in a world fixed coordinate frame W is defined. Usually the size is chosen equal to the maximum measurement distance of the radar. The initial orientation is either the same as the ego vehicles orientation at that time or taken from a memory. The size of each cell is also defined. In Algorithm 2.1 the cell size 1 is chosen in order to simplify the succeeding equations. For a normalized cell size float values, such as distances, can simply be rounded and the integer values are used. The initial log odds of all map cells is defined, as well as the amount of evidence a measurement carries for occupied or free cells.

The map's position is updated in the the second step. Introducing a map coordinate frame M, with the same orientation as the world frame W and with its origin O^M at the center of the middle cell of the map. The ego vehicles origin O^E, lies in the same cell as O^M, but not necessarily in its center. The introduction of M is required in order to tackle problems arising from the discretization of the grid map. As the ego vehicle moves new rows and columns of the map are initiated and rows and columns corresponding to negotiated areas are removed. The map size is kept constant.

In step 4 the log odds of the cells containing an observation is increased, cf. (9). The value by which the log odds is increased is inversely proportional to the measured range, as it is assumed that the accuracy of a measurement is dependent on the measured range. The log odds of the cell underlaying a straight line between the ego vehicle and the observation is reduced in step 5. This is accomplished by iterating the grids of one of the map frame's axes and calculating the corresponding index on the orthogonal axis, i.e., finding the grid cell crossed by the radar beam.

[1] This work was carried out together with Volvo Car Corporation and the Intelligent Vehicle Safety System Program (IVSS). The results were validated using a Volvo S80 equipped with a ACC3 77 GHz mechanically scanned FMCW radar and a forward looking camera.

Algorithm 2.1 (Occupancy Grid Mapping)

1. **Initialize** *the log odds matrix* $\ell \in \mathbb{R}^{N_{OGM} \times N_{OGM}}$ *representing the map with e.g.,*

$$\ell_0 = 0.$$

and the position and orientation the map

$$\begin{bmatrix} x^W_{O^M O^W,0} & y^W_{O^M O^W,0} \end{bmatrix}^T = \begin{bmatrix} 0 & 0 \end{bmatrix}^T.$$

Define the cell size. Here it is assumed that the cell size is normalized to 1, which simplifies the calculations. Choose thresholds $\ell_{occ} > \ell_0$ *and* $\ell_{free} < \ell_0$.

2. **Update map position** *with the ego vehicles movement*

$$\texttt{shift = round(}d^W_{O^E O^W,t}\texttt{-}d^W_{O^M O^W,t-1}\texttt{)}$$

which is used to remove past rows or columns and initiate new

```
if shift(1)>=0
    ℓ = [ℓ(:,shift(1)+1:N_OGM),ℓ_0*ones(N_OGM,shift(1))];
else
    ℓ = [ℓ_0*ones(N_OGM,-shift(1)),ℓ(:,1:N_OGM+shift(1))];
end
if shift(2)>=0
    ℓ = [ℓ_0*ones(shift(2),N_OGM);ℓ(1:N_OGM-shift(2),:)];
else
    ℓ = [ℓ(-shift(2)+1:N_OGM,:);ℓ_0*ones(-shift(2),N_OGM)];
end
```

and update the origin of the map

$$d^W_{O^M O^W,t} \texttt{ = } d^W_{O^M O^W,t-1} \texttt{ + shift;}$$

3. **Transform measurements** *from polar coordinates, range* d *and bearing* $\psi_{\mathcal{S}_i O^E}$ *given in the vehicle's coordinate frame* E*, into Cartesian coordinates in the map's coordinate frame* M

$$d^E_{\mathcal{S}_i O^E} = d_i \begin{bmatrix} \cos\psi_{\mathcal{S}_i O^E} \\ \sin\psi_{\mathcal{S}_i O^E} \end{bmatrix}$$
$$d^M_{\mathcal{S}_i O^M} = A^{WE} d^E_{\mathcal{S}_i O^E} + d^W_{O^E O^W} - d^W_{O^M O^W}$$

for $i = 1, \ldots, N_m$ *where* N_m *is the number of current measurements.*

4. **Increase** *the occupancy log odds of the those cells which can be associated to the measurements* $i = 1, \ldots, N_m$*, cf.* (9)

$$\texttt{index = round(}d^M_{\mathcal{S}_i O^M}\texttt{);}$$
$$\ell_t\texttt{(index) = }\ell_{t-1}\texttt{(index) + }\ell_{occ}/d_i \texttt{ - }\ell_0$$

5. **Decrease** *the occupancy log odds of the those cells which are underlaying a straight lines between the ego vehicle and the measurements* $i = 1, \ldots, N_m$, *cf.* (9)

```
if |x^M_{S_i O^E}| > |y^M_{S_i O^E}|
    for j = 0:sign(x^M_{S_i O^E}):x^M_{S_i O^E} - sign(x^M_{S_i O^E})
        index = round([j;j*y^M_{S_i O^E} / x^M_{S_i O^E}] + d^M_{O^E O^M});
        ℓ_t(index) = ℓ_{t-1}(index) + ℓ_free/d_i - ℓ_0
    end
else
    for j = 0:sign(y^M_{S_i O^E}):y^M_{S_i O^E} - sign(y^M_{S_i O^E})
        index = round([j;j*x^M_{S_i O^E} / y^M_{S_i O^E}] + d^M_{O^E O^M});
        ℓ_t(index) = ℓ_{t-1}(index) + ℓ_free/d_i - ℓ_0
    end
end
```

6. **Repeat** *the algorithm. Let* $t := t+1$, *collect new measurements, estimate the ego vehicles position and repeat from step 2.*

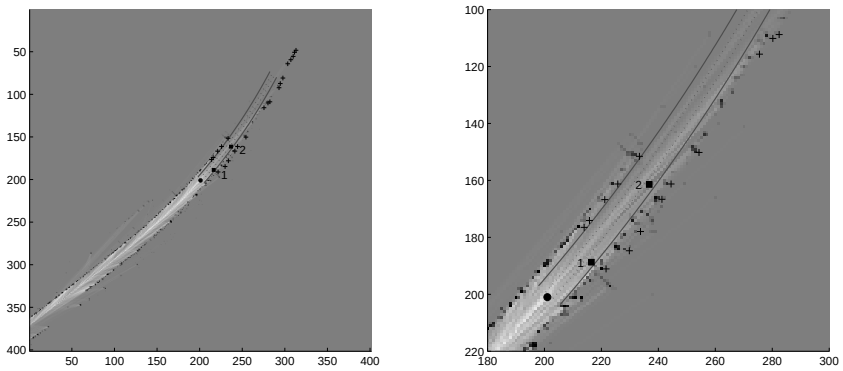
2.2.2 Experiments and Results

Figure 3a shows an OGM example of the highway situation given in the ego vehicle's camera view in Figure 3c. The size of the OGM is N_{ogm} =401 m, with the ego vehicle in the middle cell. Each cell is a 1×1 m square. The gray-level in the occupancy map indicates the probability of occupancy $p(\mathbf{m}|\mathbf{y}_{1:t}, \mathbf{x}_{E,1:t})$, the darker the grid cell, the more likely it is to be occupied. The map shows all major structural elements as they were visible at the height of the radar. This is a problem if the road is undulated and especially if the radar observes obstacles over and behind the guard rail. In this case the occupancy probability of a cell might be decreased even though it was previously believed to be occupied, since the cell lays between the ego vehicle and the new observation. The impact of this problem can be reduced by choosing ℓ_{occ} and ℓ_{free} well.

2.3 Summary of OGM

It is clearly visible in Figure 3a that the left border is sharper than the right. The only obstacle on the left side is the guardrail which gives rise to the sharp edge, whereas on the right side there are several obstacles behind the guardrail which also cause reflections, e.g. noise protection walls and vegetation. If we look closer in Figure 3b we realize that there is no black line of occupied cells representing the guard rail as expected. Instead there is a region with mixed probability of occupancy and after about 5 m the gray region with initial valued cells tells us that we know nothing about these cells.

In summary the OGM generates a good-looking overview of the traffic situation, but not much information for a collision avoidance system. It is inefficient



(a) (b)

(c)

Figure 3: The filled circle at position $(201, 201)$ in the occupancy grid map in Figure (a) is the ego vehicle, the stars are the radar observations obtained at this time sample, the black squares with numbers 1 and 2 are the two leading vehicles that are currently tracked. The gray-level in the figure indicates the probability of occupancy, the darker the grid cell, the more likely it is to be occupied. The shape of the road is given as solid and dashed lines, calculated as described in [25]. The camera view from the ego vehicle is shown in Figure (c), the concrete walls, the guardrail and the pillar of the bridge are interesting landmarks. Furthermore, the two tracked leading vehicles are clearly visible in the right lane.

to represent the occupancy information as a rather huge square matrix with most of its elements equal 0.5, which indicates that we don't know anything about these cells. A alternative way to represent the result is discussed in the next section.

3 Curve Fitting Using a Quadratic Program

The key in this approach is to make use of the road curvature estimate from the sensor fusion framework [25] mentioned in the introduction to sort the stationary radar measurements according to which side of the road they originate from. These measurements are then used together with the estimates from the sensor fusion to dynamically form a suitable constrained quadratic program (QP) for estimating the free space in front of the vehicle. This QP models the temporal correlation that exists in roads and the fact that the road shape cannot change arbitrarily fast.

3.1 Problem Formulation

A stationary object m will be referred to as an observation in the point $\mathcal{S}_m$. The radar in the ego vehicle measures the azimuth angle $\psi^E_{\mathcal{S}_m O^E}$ and the range $d = ||d^E_{\mathcal{S}_m O^E}||_2$ to the stationary object. These are transformed into Cartesian coordinates $\begin{bmatrix} x_{\mathcal{S}_m O^E} & y_{\mathcal{S}_m O^E} \end{bmatrix}^T$ in any coordinate frame.

All the observations of stationary objects $\mathrm{S} = \{\mathcal{S}_m\}_{m=1}^{N_\mathcal{S}}$ from the radar are sorted into two ordered sets, one for the left side $\mathrm{S_l}$ and one for the right side $\mathrm{S_r}$ of the road. In order to be able to perform this sorting we need some information about the road geometry, otherwise it is of course impossible. The information about the road shape in (1) can be used to decide if an observation should be sorted into the left set according to

$$\mathrm{S_l} = \left\{ \mathcal{S}_m \in \mathcal{S} \mid y^E_{\mathcal{S}_m O^E} \geq l + \delta_r x^E_{\mathcal{S}_m O^E} + \frac{c_0}{2} (x^E_{\mathcal{S}_m O^E})^2 \right\} \tag{13}$$

or the right set according to

$$\mathrm{S_r} = \left\{ \mathcal{S}_m \in \mathcal{S} \mid y^E_{\mathcal{S}_m O^E} < l + \delta_r x^E_{\mathcal{S}_m O^E} + \frac{c_0}{2} (x^E_{\mathcal{S}_m O^E})^2 \right\}. \tag{14}$$

Observations which lay more than 200 m behind the vehicle are removed from the set. The two sets $\mathrm{S_l}$ and $\mathrm{S_r}$ are resorted at every sample, according to the new curvature estimate.

Given the data in $\mathrm{S_l}$ we seek a road border model, provided by a predictor

$$\hat{y}^E_{\mathcal{S}_m O^E}(x^E_{\mathcal{S}_m O^E}, \theta), \tag{15}$$

where θ denotes a parameter vector describing the road borders. The exact form of this predictor is introduced in the subsequent section, where two different predictors are derived. The data in $\mathrm{S_r}$ is treated analogously. The road border parameters θ are estimated by solving the following least-square problem

$$\min_\theta \quad \sum_{i=1}^{N_{\mathrm{S_l}}} \lambda_i \left(y^E_{\mathcal{S}_m O^E} - \hat{y}^E_{\mathcal{S}_m O^E}(x^E_{\mathcal{S}_m O^E}, \theta) \right)^2, \tag{16}$$

where $N_{\mathrm{S_l}}$ is the number of observations and λ_i is a weighting factor. The problem (16) is formulated as if there is only an error in the y-coordinate. Obviously there are errors present also in the x-coordinate. This can be taken care of by formulating a so called errors-in-variables problem (within the optimization literature this problem is referred to as a total least squares problem), see e.g., [3].

However, for the sake of simplicity we have chosen to stick to an ordinary least squares formulation in this work. This might also be justified by the radar having less range uncertainty than azimuth.

3.2 Road Border Model

In this section we will derive and analyze two different predictor models, one linear and one nonlinear.

An important problem to be solved is to decide which radar measurements should be used in estimating the parameters. Later in this section we will introduce suitable constraints that must be satisfied. This will allow us to remove non-relevant data, i.e., outliers.

3.2.1 Predictor

The two ordered sets $\mathrm{S_l}$ and $\mathrm{S_r}$ are handled analogously. Hence, only the processing related to the left set is described here. The observations are expressed in the world coordinate frame W when they are stored in $\mathrm{S_l}$. Obviously it is straightforward to transform them into the vehicle's coordinate frame E, using the rotation matrix $R^{EW} = (R^{WE})^T$.

As depicted earlier the lanes are modeled using the polynomial (1). Let us assume that the white lane markings are approximately parallel to the road border. In order to allow the number of lanes to change, without simultaneously changing the curvature, we extend the second order model (1) with a fourth element. Hence, a linear predictor is provided by

$$\hat{y}_1^E(x^E, \theta_{1,l}) = l_0 + l_1 x^E + l_2 (x^E)^2 + l_3 (x^E)^3, \tag{17}$$

which is a third order polynomial, describing the road's left border, given in the ego vehicle coordinate frame E.

By analyzing road construction standards, such as [41], we assume that the increment and decrement of the number of lanes can be modeled using the arctan function illustrated in Figure 4a. This allows for a continuous, but possible rapid, change in shape. Let us now, as a second approach, extend (1) and form the following nonlinear predictor

$$\hat{y}_2^E(x^E, \theta_{2,l}) = l_0 + l_1 x^E + l_2 (x^E)^2 + k \arctan \tau (x^E - b), \tag{18}$$

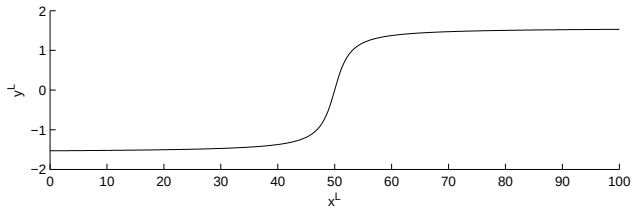
where the parameter b indicates where arctan crosses zero. The slope τ and magnitude k could be chosen according to typical road construction constants. An example of the complete nonlinear road border model (18) is shown in Figure 4b.

We will start describing the linear model (17) and come back to the nonlinear model (18) later in this section. Given the N_{S_l} observations in $\mathrm{S_l}$, the parameters

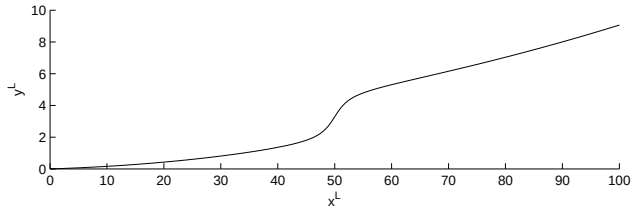
$$\theta_{1,l} = \begin{bmatrix} l_0 & l_1 & l_2 & l_3 \end{bmatrix}^T \tag{19}$$

can be approximated by rewriting the linear predictor (17) according to

$$\widehat{Y}_1^E = (\Phi^E)^T \theta_{1,l}, \tag{20}$$



(a)



(b)

Figure 4: A pure arctan is shown in Figure (a), whereas the complete expression (18) is shown in Figure (b) for a typical example.

where the regressors ($i = 1, \ldots, N_{\mathrm{S}_l}$)

$$\varphi_i^E = \begin{bmatrix} 1 & x^E_{\mathcal{S}_m O^E} & (x^E_{\mathcal{S}_m O^E})^2 & (x^E_{\mathcal{S}_m O^E})^3 \end{bmatrix}^T. \tag{21}$$

are stacked on top of each other in order to form

$$\Phi^E = \left[\varphi_1^E, \ldots, \varphi_{N_{\mathrm{S}_l}}^E \right], \tag{22}$$

The parameters are found by minimizing the weighted least square error (16), here in matrix form

$$||Y^E - \Phi^E \, \theta_{1,l}||^2_\Lambda = (Y^E - \Phi^E \, \theta_{1,l})^T \Lambda (Y^E - \Phi^E \, \theta_{1,l}), \tag{23}$$

where Λ is a weighting matrix

$$\Lambda_l = \mathrm{diag}(\begin{bmatrix} \lambda_1 & \cdots & \lambda_{N_{\mathrm{S}_l}} \end{bmatrix}). \tag{24}$$

and the y-coordinates are given by

$$Y^E = \left[y_1^E, \ldots, y_{N_{\mathrm{S}_l}}^E \right]^T. \tag{25}$$

The right hand side of the road is modeled analogously, using the following parameter vector,

$$\theta_{1,r} = \begin{bmatrix} r_0 & r_1 & r_2 & r_3 \end{bmatrix}^T. \tag{26}$$

The azimuth angle $\psi_{\mathcal{S}_m O^E}$ is measured with lower accuracy than the range d in the radar system. This influences the uncertainty of the measurements, when transformed into Cartesian coordinates in accordance to the measured distance. Therefore, the elements of the weight matrix Λ_l in (23) are defined as

$$\lambda_i = \frac{1}{\log d_i}, \quad i = 1, \ldots, N_{\mathrm{S}_l}, \tag{27}$$

modeling the fact that stationary objects close to the vehicle are measured with higher accuracy than distant objects. Hence, the closer the object is, the higher the weight.

The problem of minimizing (23) can be rewritten as a quadratic program [4] according to

$$\min_{\theta_{1,l}} \quad \theta_{1,l}^T \Phi^T \Lambda_l \Phi \theta_{1,l} - 2(Y^E)^T \Lambda_l \Phi \theta_{1,l}. \tag{28}$$

A straightforward solution of this problem will not work due to the simple fact that not all of the stationary objects detected by the radar stems from relevant objects for our purposes. For example, under some circumstances the radar also detects objects at the opposite side of the highway. These observations could for example stem from a guard rail or the concrete wall of a gateway from e.g. a bridge, see Figure 5b for an example. If the road borders are estimated according to the quadratic program in (28) using these observations the result will inevitably be wrong. In order to illustrate that this is indeed the case the result is provided in Figure 5a. In the subsequent section we will explain how this situation can be avoided by deriving a set of feasibility conditions that the curve parameters θ_l and θ_r have to fulfill.

Let us briefly revisit the nonlinear model (18). Since this predictor is nonlinear, it cannot be factored in the same way as we did for the linear predictor in (20). Instead, we have to keep the nonlinear form, resulting in the following optimization problem to be solved

$$\min_{\theta_{2,l}} \quad \left\| Y^E - \widehat{Y}_2^E(X^E, \theta_{2,l}) \right\|_{\Lambda_l}^2, \tag{29}$$

where Y^E was defined in (25) and similarly $\widehat{Y}_2^E$ are the nonlinear predictions

$$\hat{y}_2^E(x^E, \theta_{2,l}) = l_0 + l_1 x^E + l_2 (x^E)^2 + k \arctan \tau (x^E - b) \tag{30}$$

stacked on top of each other. Hence, the parameters $\theta_{2,l}$ used in (29) are given by

$$\theta_{2,l} = \begin{bmatrix} l_0 & l_1 & l_2 & k & \tau & b \end{bmatrix}^T. \tag{31}$$

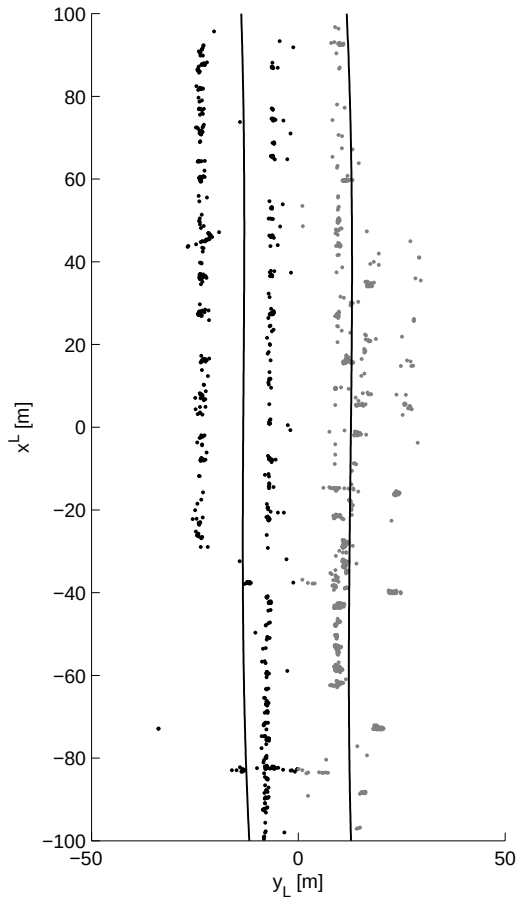
The resulting problem (29) is a non-convex least-squares problem.

3.2.2 Constraining the Predictor

The predictor has to be constrained for the problem formulation to be interesting. More specifically, we will in this section derive constraints forming a convex set, guaranteeing that the resulting linear optimization problem remains quadratic. This problem can then be efficiently solved using a dual active set method[2] [15].

As we assume that the white lane markings (1) are approximately parallel to the road border (17), we could use the angle δ_r to constrain the second border

[2]The QP code was provided by Dr. Adrian Wills at the University of Newcastle, Australia, see http://sigpromu.org/quadprog. This code implements the method described in [16, 33].



(a)

(b)

Figure 5: The gateway shown on the opposite side of the highway in Figure (b) misleads the road boarder estimation. The stored observations are shown together with the estimated road borders (lines) in Figure (a). The black points belongs to the left set S_l and the gray points belongs to the right set S_r.

parameter l_1 and we could use the curvature c_0 to constrain the third border parameter l_2 according to

$$(1-\Delta)\delta_r - \epsilon_{\delta_r} \leq l_1 \leq (1+\Delta)\delta_r + \epsilon_{\delta_r} \quad \text{if } \delta_r \geq 0, \tag{32a}$$
$$(1+\Delta)\delta_r - \epsilon_{\delta_r} \leq l_1 \leq (1-\Delta)\delta_r + \epsilon_{\delta_r} \quad \text{if } \delta_r < 0, \tag{32b}$$
$$\frac{(1-\Delta)c_0 - \epsilon_{c_0}}{2} \leq l_2 \leq \frac{(1+\Delta)c_0 + \epsilon_{c_0}}{2} \quad \text{if } c_0 \geq 0, \tag{32c}$$
$$\frac{(1+\Delta)c_0 - \epsilon_{c_0}}{2} \leq l_2 \leq \frac{(1-\Delta)c_0 + \epsilon_{c_0}}{2} \quad \text{if } c_0 < 0, \tag{32d}$$

where the allowed deviation Δ is chosen as 10%, i.e., $\Delta = 0.1$. A small value ϵ is added to avoid that both the upper and lower bounds are equal to 0 in case δ_r or c_0 is equal to 0. Several different approaches for estimating the road curvature c_0 are described in [26].

The first border parameter l_0 is not constrained, because the number of lanes may change at e.g. a gateway. It should be possible for the border of the road to move in parallel to the ego vehicles motion without any conditions.

In order to create a feasibility condition for the fourth parameter l_3 of the linear model, the estimated position of the ego vehicle expressed in the world frame W is saved at each time sample. A data entry is removed from the set if it lays more than 200 m behind the current position. Furthermore, the estimated curvature is used to extrapolate points 200 m in front of the vehicle. These points together with information about the ego vehicle's earlier positions are used to derive a driven path as a third order polynomial

$$y^E = l + \delta_r x^E + \frac{c_0}{2}(x^E)^2 + \frac{c_1}{6}(x^E)^3. \tag{33}$$

Especially the parameter c_1 is of interest and can be used to constrain l_3. Hence, the final inequality, which will further constrain (28) is given by

$$\frac{(1-\Delta)c_1 - \epsilon_{c_1}}{6} \leq l_3 \leq \frac{(1+\Delta)c_1 + \epsilon_{c_1}}{6} \quad \text{if } c_1 \geq 0, \tag{34a}$$
$$\frac{(1+\Delta)c_1 - \epsilon_{c_1}}{6} \leq l_3 \leq \frac{(1-\Delta)c_1 + \epsilon_{c_1}}{6} \quad \text{if } c_1 < 0. \tag{34b}$$

To summarize, the constrained optimization problem to be solved based on the linear predictor (17) is given by

$$\begin{aligned} \min_{\theta_{1,l}} \quad & \|Y^E - \widehat{Y}_1^E(X^E, \theta_{1,l})\|_{\Lambda_l}^2 \\ \text{s.t.} \quad & (32) \\ & (34) \end{aligned} \tag{35}$$

and the implementation steps are given in Algorithm 3.1.

Algorithm 3.1 (Linear Model)

We are only showing the steps for the left set, the right set i handled analogously.

1. **Resort** *the set* `S_l_ego`*, expressed in the ego vehicle frame, into* $\begin{bmatrix}\Phi^T & Y^E\end{bmatrix}$

```
S_l_ego = [ones(size(S_l_ego,2),1), S_l_ego(1,:),...
           S_l_ego(1,:).^2, S_l_ego(1,:).^3, S_l_ego(2,:)];
```

   cf. (21), (22) *and* (25).

2. **Set constraints** *according to* (32) *and* (34)*, i.e., set the upper and lower bounds,* `ub` *and* `lb`*, respectively. The bound on* l_0 *is infinite*

```
lb(1)=-inf;
ub(1)=inf;
```

the bounds for l_1 *and* l_2 *are given in* (32)*. The steps to derive the bound for* l_3 *are more complicated and repeated here for clarity. Start by adding the actual ego vehicle position (*$d^W_{O^E O^W}$ =`x_ego, y_ego`*) to the driven path* `map.path` *and remove objects laying more than* 100 m *behind*

```
path = [x_ego, y_ego; map.path];
path_index = find(sqrt((path(:,1)-x_ego).^2...
    +(path(:,2)-y_ego).^2)<100);
map.path = [];
map.path = path(path_index,:);
```

The positions of the driven path are merged with the predicted path to derive the constraint for l_3

```
x_path = [1:100];
y_path = δr*x+c0/2*x.^2;
path_ego = [[x_path',y_path'];...
    (R^EW*(map.path-ones(size(index_path))*[x_ego,y_ego])')'];
path_ego = [path_ego(:,1), path_ego(:,1).^2,...
    path_ego(:,1).^3, path_ego(:,2)];
path_coor = path_ego(:,1:3)\path_ego(:,4);
if path_coor(3)>=0
    lb(4) = (0.9*path_coor(3)-1e-5)/6;
    ub(4) = (1.1*path_coor(3)+1e-5)/6;
else
    ub(4) = (0.9*path_coor(3)+1e-5)/6;
    lb(4) = (1.1*path_coor(3)-1e-5)/6;
end
```

where the last if clause is given in (34)*.*

3. **Solve** *the quadratic program, with weighting matrix* Λ_l *according to* (24)*.*

```
Λl = diag(1./log(S_l(3,:)))
l_coor = qpas(S_l_ego(:,1:4)'*Λl*S_l_ego(:,1:4),...
    -(S_l_ego(:,5)'*Λl*S_l_ego(:,1:4))',...
    [],[],[],[],lb',ub');
```

where the third row of the set `(S_l(3,:))` *is the measured range.*

4. **Euclidean distance** *from all points in the set to the line is derived to be used for outlier rejection*

```
l_dist=S_l_ego(:,5)-(l_coor'*S_l_ego(:,1:4)')';
```

The parameter b, of the nonlinear model (18) is constrained by the measurement distance and the parameters k and τ are constrained by road construction standards. The resulting nonlinear least-squares problem is finally given by

$$\begin{aligned} \min_{\theta_{2,l}} \quad & \|Y^E - \widehat{Y}_2^E(X^E, \theta_{2,l})\|_{\Lambda_l}^2 \\ \text{s.t.} \quad & (32) \\ & b_{\max} \leq b \leq -b_{\max} \\ & k_{\max} \leq k \leq -k_{\max} \\ & \tau_{\max} \leq \tau \leq \tau_{\min}. \end{aligned} \tag{36}$$

and the implementation steps are given in Algorithm 3.2.

Algorithm 3.2 (Nonlinear Model)

We are only showing the steps for the left set, the right set i handled analogously.

1. **Set constraints** *according to* (32) *and* (36)*, i.e., set the upper and lower bounds,* `ub` *and* `lb`*, respectively.*
2. **Solve** *the nonlinear least-square problem*

```
l_V = @(x) (S_l_ego(2,:)-x(1)-S_l_ego(1,:).*x(2)...
    -S_l_ego(1,:).^2.*x(3)/2...
    -x(4).*atan(x(5).*(S_l_ego(1,:)-x(6))))'*(S_l_ego(2,:)...
    -x(1)-S_l_ego(1,:).*x(2)-S_l_ego(1,:).^2.*x(3)/2...
    -x(4).*atan(x(5).*(S_l_ego(1,:)-x(6))));
l_coor = fmincon(l_V,l_coor,[],[],[],[],lb',ub',[],options);
```

   where `S_l_ego(:,i)` *is* $\begin{bmatrix} x^E_{S_i O^E} & x^E_{S_i O^E} \end{bmatrix}$.
3. **Euclidean distance** *from all points in the set to the line is derived to be used for outlier rejection*

```
l_dist=(S_l_ego(2,:)-l_coor(1)-S_l_ego(1,:).*l_coor(2)...
    -S_l_ego(1,:).^2.*l_coor(3)/2...
    -l_coor(4).*atan(l_coor(5).*(S_l_ego(1,:)-l_coor(6))))';
```

3.2.3 Outlier Rejection

The difference between the observed point and the calculated road border lines is used to separate and remove outliers which lie more than 1.5 lane width (W) from the lines. Subsequently the quadratic program (28) is used a second time and the result is shown in Figure 6. For this case, the two predictor models yields approximately the same result.

An advantage of the nonlinear model is its ability to model changes in the number of lanes, as can be seen in Figure 7a, where the number of lanes changes from two to three. Recall that it is the use of the arctan function that allows us to model changes in the number of lanes. The new lane originates from an access road to the highway. The corresponding camera view is shown in Figure 7b.

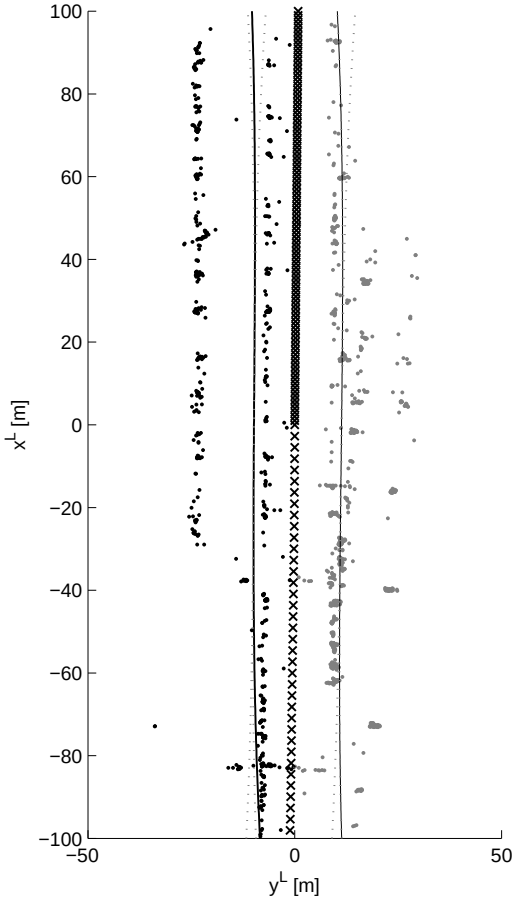
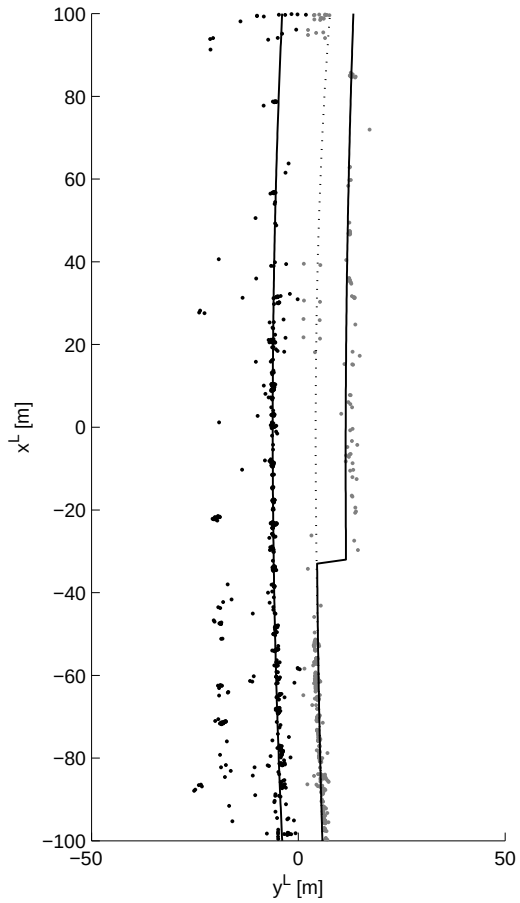


Figure 6: Road border estimation for the same situation as in Figure 5a, but the additional constraints are now used. The feasible set for the parameters l_1, l_2 and l_3 is between the dashed lines. The crosses shows the driven path (for $x < 0$) and the estimated path (for $x > 0$).

3.2.4 Computational Time

We have compared the computation time for the two proposed predictors with constraints. The nonlinear least square problem (36) was solved using the function `fmincon` in MATLABS optimization toolbox. Furthermore, we have used two different methods in solving the quadratic problem (35). The first method is the active set method mentioned earlier, where parts are written in C-code, and the second method used is `quadprog` in MATLABS optimization toolbox. The computational time was averaged over a sequence of 1796 samples. The sample time is 0.1 s, implying that the measurements were collected during 179.6 s highway driving. The results are shown in Table 2.

The computation time of the nonlinear predictor is about 38 % higher than



(a)

(b)

Figure 7: A change in the number of lanes is modeled accurately using the arctan function in the nonlinear predictor, as shown by the solid line in Figure (a). The dashed line is the result of the linear predictor. The camera view of the traffic situation is shown is Figure (b).

it is for the linear predictor proposed in this paper. The MATLAB function `quadprog` needs 149 % more computational time. This indicates that the computational time of the nonlinerar predictor can possibly be reduced by utilizing an optimized C-code implementation.

Table 2: Average computational time for one sample.

Method	Time [ms]
Linear Predictor (this paper)	84
Linear Predictor (`quadprog`)	209
Nonlinear Predictor	116

3.3 Calculating the Free Space

The free distance to the left and the right road borders is now easily calculated by considering the first parameters l_0 and r_0 respectively. The number of lanes on the left hand side is given by

$$\max\left(\left\lfloor\frac{l_0 - l}{w}\right\rfloor, 0\right) \tag{37}$$

and the number of lanes on the right hand side is given by

$$\max\left(\left\lfloor\frac{-r_0 - r - 2}{w}\right\rfloor, 0\right). \tag{38}$$

In the expressions above l and r are the distances from the sensor in the ego vehicle to the left and right lane markings of the currently driven lane. We assume that the emergency lane is 2 m on the right hand side of the road [41].

The number of observed stationary objects depends on the surrounding environment. A guard rail or a concrete wall results in more observations than for example a forest. Hence, the estimated border lines are accompanied by a quality measure which depends on the number of observations and their variance. The variance is calculated before and after the outliers have been removed.

It is still a problem to detect the distance to the road border if there is a noise protection wall some meters to the right of the road. This wall generates many observations with small variance and cannot be distinguished from a guard rail. However, one solution might be to include camera information in a sensor fusion framework.

3.3.1 Border Line Validity

A very thrilling problem with the present curve fitting approach is that there are no gaps to properly leave or enter the road at a gateway. A collision avoidance system would brake the vehicle automatically if leaving the road at a gateway when simultaneously crossing the border line. This leads us to the conclusion that the border lines should only be defined if the number of observations around it lies above a certain limit.

In a first step we calculate the distance between the line and the observations in the set $\mathrm{S_l}$

$$d_{l,i} = \left| y_i^E - \left(\delta_r x_i^E + \frac{c_0}{2}(x_i^E)^2\right)\right| \quad \text{for} \quad i = 1, \ldots, N_{\mathrm{S_l}} \tag{39}$$

and compare it with a constant or variable, e.g. the lane width w

$$n_i = \begin{cases} 1 & \text{if } d_{l,i} > w \\ 0 & \text{otherwise.} \end{cases} \tag{40}$$

In a second step the border line is segmented into valid and not valid parts. The start and end points of the valid parts are given by identifying the indices of two non-equal and adjoined elements in the vector n. By applying the XOR function ($\oplus$) according to

$$c = n_{2:n_{S_l}} \oplus n_{1:n_{S_l}-1}, \tag{41}$$

the start and end points of the border line are identified as the indices with $c = 1$. These indices are stored in two additional sets for the left and right border lines, respectively. An example is shown in Figure 8a and the corresponding camera view in Figure 8b. The gateway to the right leads to a gap in the right border line, between $48 - 73$ m ahead of the ego vehicle. One of the leading vehicles lies between the ego vehicle and the guard rail, this is the reason why there are so few stationary object on the left hand side from about 70 m ahead and why no line could be drawn.

3.3.2 Summary of Curve Fitting using QP

This approach fuses the information from several radar measurements. It sorts out the featureless information and interpolates the road borders as an optimization problem. The main steps introduced in this section are repeated and summarized in Algorithm 3.3.

Algorithm 3.3 (Curve Fitting using QP)

1. **Transform** *sets from world to ego vehicle coordinate frame and merge the left and right set into one matrix of size* $\mathbb{R}^{2\times(N_{S_l}+N_{S_r})}$

   $$\texttt{S\_ego} = R^{EW}\texttt{[S}_\texttt{l}\texttt{,S}_\texttt{r}\texttt{]} + d^W_{O^E O^W}\texttt{;}$$
   $$\texttt{S} = \texttt{[S}_\texttt{l}\texttt{,S}_\texttt{r}\texttt{];}$$

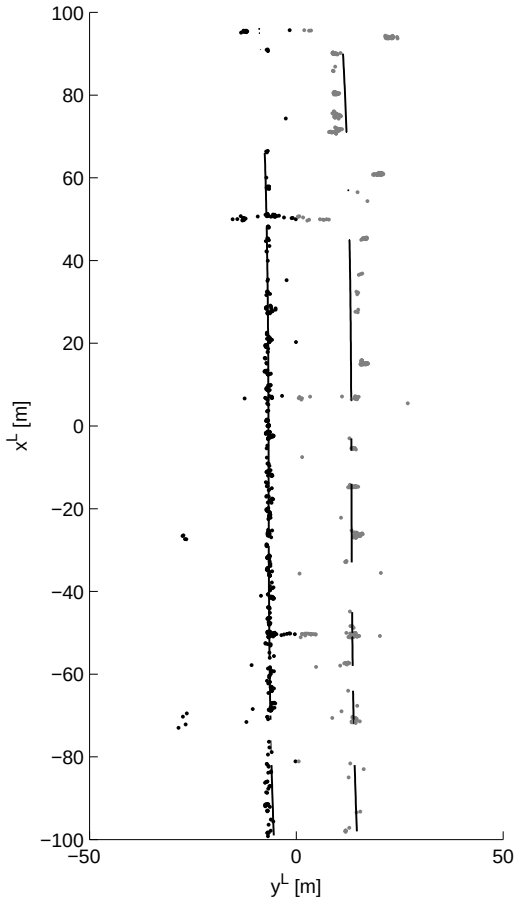
2. **Remove** *data, which lies outside the area of interest, from the sets. This is usually data laying behind the vehicle.*

3. **Resort sets** *according to the new road curvature estimate* c_0

   ```
   S_index_l = find(S_ego(2,:)  > c_0*S_ego(1,:).^2/2);
   S_index_r = find(S_ego(2,:)  <= c_0*S_ego(1,:).^2/2);
   S_l = S(:,S_index_l);
   S_r = S(:,S_index_r);
   ```

4. **Add new measurements** *to set. Start by transforming polar coordinates (range* d *and bearing* $\psi_{\mathcal{S}_m O^E}$*) into Cartesian coordinates*

   $$d^E_{\mathcal{S}_m O^E} = d_m \begin{bmatrix} \cos\psi_{\mathcal{S}_m E} \\ \sin\psi_{\mathcal{S}_m O^E} \end{bmatrix}$$
   $$d^W_{\mathcal{S}_m O^W} = R^{WE} d^E_{\mathcal{S}_m O^E} + d^W_{O^E O^W}$$



(a)

(b)

Figure 8: The gateway to the right in Figure (b) leads to a gap in the right border line, between $48 - 73$ m ahead, as shown in Figure (a).

for $m = 1, \ldots, N_S$ where N_S is the number of current received measurements. After this, decide if the measurement shall be sorted into the left or right set

$$\texttt{if } y^W_{S_m O^W} > \frac{c_0}{2}\left(x^W_{S_m O^W}\right)^2$$
$$\texttt{S\_l = [S\_l,[}d^W_{S_m O^W}\texttt{;}d_m\texttt{]];}$$
$$\texttt{else}$$
$$\texttt{S\_r = [S\_r,[}d^W_{S_m O^W}\texttt{;}d_m\texttt{]];}$$
$$\texttt{end}$$

where the third column is the measured distance which is stored in the set to be used as weighting factor.

5. **Transform** *sets into ego vehicle coordinate frame, cf. step 1.*
6. **Optimization** *is described in Algorithms 3.1 and 3.2 for the linear and nonlinear models, respectively.*
7. **Remove outliers** *laying to far from the expected line*

```
l_index = find(abs(l_dist)<1.5*w);
S_l_ego_new = S_l_ego(:,l_index);
```

for both left and right set, in this example. A point is defined as an outlier if it lies more than 1.5 times the road width w from the border line.

8. **Optimization** *algorithm is run again, but this time without outliers. The implementation is given as step 3-4 in Algorithm 3.1 for the linear model and as step 2-3 in Algorithm 3.2 for the nonlinear model.*
9. **Border line validity** *segments the line into valid and not valid parts. Start by sorting the stack expressed in the ego coordinate frame with respect to its x-coordinate. Use the Euclidean distance (39) between all points in the set and the border line in (40) to decide if a point is valid or not, and use (41) do find switches between valid and not valid segments. By analyzing the first point it is then possible to decide whether the switches are start or end points, hence, if the segment is valid or not.*
10. **Repeat** *from step 1.*

The main advantage of this method is the simple and convenient representation in the form of two polynomials with four to six parameters, depending on the chosen method, describing the road borders to the left and to the right as two lines. These parameters are accomplished with vectors describing the start- and end-points of valid segments of these lines, thereby allowing gaps at e.g. gateways.

The drawback of this approach is that it is rather unstable if there are too few measurement points or if the measurements stems from other objects than the guardrail. However, the problem of having too few measurements or having measurements from the wrong objects is very hard to solve with any algorithm.

4 Tracking Stationary Extended Objects for Road Mapping

In this section we consider the problem of estimating the position and shape of stationary objects in front of the vehicle. We represent the stationary objects as

- points, with sources such as delineators or lampposts or
- lines, where measurements stem from e.g. guard rails or concrete walls.

The lines are modeled as extended objects, since an object is denoted extended whenever the object extent is larger than the sensor resolution. Put in other words, if an object should be classified as extended does not only depend on its physical size, but also on the physical size relative to the sensor resolution. Extended object tracking is extensively described in e.g. [35, 14] and it has got quite recent attention in [40, 1] where Monte Carlo methods are applied and in [22] which is based on random matrices.

The stationary objects are tracked by casting the problem within a standard sensor fusion framework. Since we use a linear model and assume Gaussian noise we use the standard Kalman filter [19].

4.1 Extended Object Model

In this section we introduce models for the tracked stationary object, i.e., points and lines. To take care of the lines a model with the object's shape, size, position and orientation is introduced.

4.1.1 Motion Model of the Stationary Objects

Stationary objects are modeled as points $\mathcal{P}$ or lines $\mathcal{L}$. A point $\mathcal{P}_i$ is represented using a position in the planar world coordinate frame W. Hence, the state of a point is simply

$$\mathbf{x}_{\mathcal{P}_i} \triangleq \begin{bmatrix} x^W_{\mathcal{P}_i O^W} & y^W_{\mathcal{P}_i O^W} \end{bmatrix}^T. \tag{43}$$

A line $\mathcal{L}_j$ is represented as a second order polynomial in its coordinate frame L_j

$$y^{L_j} = a_0 + a_1 x^{L_j} + a_2 \left(x^{L_j}\right)^2. \tag{44}$$

The coordinate frame L_j is initiated together with the line and is equal to the ego vehicles coordinate frame at the moment the line is created. Unlike the ego vehicle's frame it is fixed to the world system, i.e., $d^W_{O^{L_j} O^W}$ and $\psi_{L_j W}$ are constant and it does not follow the vehicles movement.

The state of the line $\mathcal{L}_j$ is

$$\mathbf{x}_{\mathcal{L}_j} = \begin{bmatrix} a_{0,j} & a_{1,j} & a_{2,j} & s_j & e_j \end{bmatrix}^T, \tag{45}$$

where s_j and e_j are the start and end points of the line given as scalar x^{L_j} values.

The motion model of the stationary objects in the form

$$\mathbf{x}_{t+1} = F\mathbf{x}_t + w_t, \quad w_t \sim \mathcal{N}(0, Q), \tag{46}$$

is simple, since the objects are not moving. For the points, the system matrix, referred to as $F_{\mathcal{P}}$, is the identity matrix. The term w_t in (46) represents the process noise. We include some dynamics into the motion model of the line. We assume that the lines are shrinking with a factor $\lambda < 1$ according to

$$s_{j,t+1} = s_{j,t} + \lambda(e_{j,t} - s_{j,t}), \tag{47a}$$

$$e^{j}_{j,t+1} = e_{j,t} - \lambda(e_{j,t} - s_{j,t}), \tag{47b}$$

leading to

$$F_{\mathcal{L}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1-\lambda & \lambda \\ 0 & 0 & 0 & \lambda & 1-\lambda \end{bmatrix}. \tag{48}$$

This shrinking behavior for the lines allows us to automatically adjust the starting and end points of the lines according to the incoming measurements.

4.1.2 Measurement Model

The measurement equation describing the measurements relation to a point $\mathcal{P}_i$ is defined as

$$\mathbf{y}_{\mathcal{P}_i,t} = \mathbf{x}_{\mathcal{P}_i,t} + v_{\mathcal{P},t}, \quad v_{\mathcal{P},t} \sim \mathcal{N}(0, R_{\mathcal{P}}), \tag{49}$$

where the output $\mathbf{y}_{\mathcal{P}_i,t} = \mathcal{S}^{W}_{m}$ is the observation m in the world coordinate associated to the i^{th} point. The term $v_{\mathcal{P},t}$ in (49) represents the measurement noise associated with the radar. The measurement equation describing the measurements relation to a line $\mathcal{L}_j$ is

$$\mathbf{y}_{\mathcal{L}_j,t} = \begin{bmatrix} 0 & 0 & 0 & h_{14} & h_{15} \\ 1 & x^{L_j}_{\mathcal{S}_m O^{L_j}} & \left(x^{L_j}_{\mathcal{S}_m O^{L_j}}\right)^2 & 0 & 0 \end{bmatrix} \mathbf{x}_{\mathcal{L}_j,t} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_{\mathcal{L},t}, \tag{50}$$

where $\mathbf{y}_{\mathcal{L}_j,t} = \mathcal{S}^{L_j}_{m}$ is the observation m in the L_j coordinate frame and associated to line $\mathcal{L}_j$. The term $v_{\mathcal{L},t} \sim \mathcal{N}(0, R_{\mathcal{L}})$ represents the measurement noise. The first row of the measurement matrix, which determines the update of the start and the end points, depends on the position of the observation in relation to the predictions of the start and the end points according to

$$\begin{bmatrix} h_{14} & h_{15} \end{bmatrix} = \begin{cases} \begin{bmatrix} 1 & 0 \end{bmatrix} & \text{if } x^{L_j}_{\mathcal{S}_m O^{L_j}} \leq s_{j,t|t-1} \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & \text{if } x^{L_j}_{\mathcal{S}_m O^{L_j}} \geq e_{j,t|t-1} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & \text{otherwise.} \end{cases} \tag{51}$$

This type of measurement where some measured quantities ($x^{L_j}_{\mathcal{S}_m O^{L_j}}$ in our case) appear as model parameters is not conventional in dynamic estimation literature and can be considered as an extension of the so-called "errors in variables" framework. In our application, this enables us to use the Kalman filter because the resulting model is linear.

4.2 Data Association and Gating

At every sampling time, the system receives a batch of $N_{\mathcal{S}}$ observations $\mathcal{S}_m$, $m = 1, \ldots, N_{\mathcal{S}}$ from the radar. These new measurements can be associated to existing tracked points $\mathcal{P}_i$, $i = 1, \ldots, N_{\mathcal{P}}$ or to tracked lines $\mathcal{L}_j$, $j = 1, \ldots, N_{\mathcal{L}}$, or a new track is initiated. The number of association events (hypotheses) is extremely large. The classical technique to reduce the number of these hypotheses is called gating [2]. We apply gating and make a nearest-neighbor type data association based on likelihood ratio tests. Other more complicated data association methods like multiple hypothesis tracking [34] or joint probabilistic data association [2] can also be used in our framework. However, these are quite complicated and computationally costly approaches and the nearest neighbor type algorithm we used has been found to give sufficient performance for our case. The gating and the data association are performed according to the following calculations. The likelihood $\ell_{\mathcal{S}_m,\mathcal{P}_i}$ that the observation $\mathcal{S}_m$ corresponds to the i^{th} point $\mathcal{P}_i$ is given by

$$\ell_{\mathcal{S}_m\mathcal{P}_i} = \begin{cases} \mathcal{N}(\mathcal{S}_m; \hat{\mathbf{y}}_{\mathcal{P}_i,t|t-1}, S_{\mathcal{P}_i}), & \text{if } \mathcal{S}_m \in \mathcal{G}_{\mathcal{P}_i} \\ 0, & \text{otherwise} \end{cases} \tag{52}$$

where $\hat{\mathbf{y}}_{\mathcal{P}_i,t|t-1}$ is the predicted measurement of the point $\mathcal{P}_i$ according to the model (49) and $S_{\mathcal{P}_i,t|t-1}$ is its covariance (innovation covariance) in the Kalman filter. The gate $\mathcal{G}_{\mathcal{P}_i}$ is defined as the region

$$\mathcal{G}_{\mathcal{P}_i} \triangleq \left\{ \mathbf{y} \,\middle|\, (\mathbf{y} - \hat{\mathbf{y}}_{\mathcal{P}_i,t|t-1})^T S^{-1}_{\mathcal{P}_i,t|t-1} (\mathbf{y} - \hat{\mathbf{y}}_{\mathcal{P}_i,t|t-1}) \leq \delta_{\mathcal{P}} \right\} \tag{53}$$

where $\delta_{\mathcal{P}}$ is the gating threshold.

The likelihood that the observation m corresponds to the j^{th} line state is derived by considering the orthogonal distance between the line and the observation. To simplify the calculations we assume that the curvature of the line is small and that the orthogonal distance can be approximated with the y-distance between the observation and the line expressed using the lines coordinate frame L_j, i.e.,

$$\epsilon_{\mathcal{S}_m\mathcal{L}_j} = y^{L_j}_{\mathcal{S}_m O^{L_j}} - \hat{y}^{L_j}_{\mathcal{S}_m O^{L_j}}, \tag{54}$$

where

$$\hat{y}^{L_j}_{\mathcal{S}_m O^{L_j}} \triangleq \begin{bmatrix} 1 & x^{L_j}_{\mathcal{S}_m O^{L_j}} & \left(x^{L_j}_{\mathcal{S}_m O^{L_j}}\right)^2 & 0 & 0 \end{bmatrix} \hat{\mathbf{x}}_{\mathcal{L}_j,t|t-1}. \tag{55}$$

The likelihood $\ell_{\mathcal{S}_m\mathcal{L}_j}$ that the observation corresponds to the j^{th} line is then given by

$$\ell_{\mathcal{S}_m\mathcal{L}_j} = \begin{cases} \mathcal{N}\left(\epsilon_{\mathcal{S}_m\mathcal{L}_j}; 0, \mathrm{E}\left(\Delta^2_{y_j}\right)\right), & \text{if } \begin{bmatrix} x^{L_j}_{\mathcal{S}_m O^{L_j}} \\ y^{L_j}_{\mathcal{S}_m O^{L_j}} \end{bmatrix} \in \mathcal{G}_{\mathcal{L}_j} \\ 0, & \text{otherwise} \end{cases} \tag{56}$$

where $y_j = y^{L_j}_{\mathcal{S}_m O^{L_j}}$ and the gate $\mathcal{G}_{\mathcal{L}_j}$ is defined as

$$\mathcal{G}_{\mathcal{L}_j} \triangleq \Bigg\{ \begin{bmatrix} x \\ y \end{bmatrix} \Bigg| \left[y - \hat{y}^{L_j}_{\mathcal{S}_m O^{L_j}} \right]^T \mathrm{E}\left(\Delta^2_{y_j}\right)^{-1} \times \left[y - \hat{y}^{L_j}_{\mathcal{S}_m O^{L_j}} \right] \leq \delta_{\mathcal{L}}, s_j - \delta_s < x < e_j + \delta_e \Bigg\}. \tag{57}$$

In (56) and (57), $\mathrm{E}(\Delta^2_{y_j})$ represents the uncertainty of the line in the y direction at the x-value $x^{L_j}_{\mathcal{S}_m O^{L_j}}$. This covariance has to be calculated in terms of the state estimate $\hat{\mathbf{x}}_{\mathcal{L}_j,t|t-1}$ and its covariance $P_{\mathcal{L}_j,t|t-1}$. This derivation can be made by first rewriting the line equation (45) with mean parameters and a deviation Δ

$$y + \Delta_y = (a_0 + \Delta_{a_0}) + (a_1 + \Delta_{a_1})x + (a_2 + \Delta_{a_2})x^2, \tag{58}$$

where the superscripts and subscripts are discarded for the sake of brevity. This gives

$$\Delta_y = \Delta_{a_0} + \Delta_{a_1} x + \Delta_{a_2} x^2. \tag{59}$$

Considering the squared expectation of this deviation, we obtain

$$\begin{aligned} \mathrm{E}(\Delta^2_y) &= \mathrm{E}(\Delta_{a_0} + \Delta_{a_1} x + \Delta_{a_2} x^2)^2 \\ &= \mathrm{E}\Big(\begin{pmatrix} 1 & x & x^2 \end{pmatrix} \begin{pmatrix} \Delta_{a_0} & \Delta_{a_1} & \Delta_{a_2} \end{pmatrix}^T \\ &\quad \times \begin{pmatrix} \Delta_{a_0} & \Delta_{a_1} & \Delta_{a_2} \end{pmatrix} \begin{pmatrix} 1 & x & x^2 \end{pmatrix}^T \Big) \\ &= \begin{pmatrix} 1 & x & x^2 \end{pmatrix} \mathrm{E}\Big(\begin{pmatrix} \Delta_{a_0} & \Delta_{a_1} & \Delta_{a_2} \end{pmatrix}^T \\ &\quad \times \begin{pmatrix} \Delta_{a_0} & \Delta_{a_1} & \Delta_{a_2} \end{pmatrix} \Big) \begin{pmatrix} 1 & x & x^2 \end{pmatrix}^T. \end{aligned} \tag{60}$$

Now, the expectation above is given by the upper-left 3×3 partition of the covariance matrix $\mathbf{P}_{\mathcal{L}_j,t|t-1}$ which we denote by $\mathbf{P}^{3\times 3}_{\mathcal{L}_j,t|t-1}$. Hence,

$$\begin{aligned} \mathrm{E}\left(\Delta^2_{y_j}\right) = \begin{pmatrix} 1 & x^{L_j}_{\mathcal{S}_m O^{L_j}} & (x^{L_j}_{\mathcal{S}_m O^{L_j}})^2 \end{pmatrix} \mathbf{P}^{3\times 3}_{\mathcal{L}_j,t|t-1} \\ \times \begin{pmatrix} 1 & x^{L_j}_{\mathcal{S}_m O^{L_j}} & (x^{L_j}_{\mathcal{S}_m O^{L_j}})^2 \end{pmatrix}^T. \end{aligned} \tag{61}$$

Having calculated the likelihood values, we form two matrices of likelihood values, one matrix $\Gamma_{\mathcal{P}} \in \mathbb{R}^{N_{\mathcal{S}} \times N_{\mathcal{P}}}$ with the combinations of observations and points, according to (52), and one matrix $\Gamma_{\mathcal{L}} \in \mathbb{R}^{N_{\mathcal{S}} \times N_{\mathcal{L}}}$ with the combinations of observations and lines, according to (56).

First we find the the maximum value of $\Gamma_{\mathcal{P}}$, and call the corresponding point state i_m and measurement m_m. Thereafter we find the maximum value of the m^{th} row, corresponding to measurement m_m of matrix $\Gamma_{\mathcal{L}}$ and call the corresponding line state j_m. The likelihood ratio denoted by $\Lambda(\mathcal{S}_m)$ is now given by

$$\Lambda(\mathcal{S}_m) \triangleq \frac{\ell_{S_m \mathcal{P}_{i_m}}}{\ell_{S_m \mathcal{L}_{j_m}}}. \tag{62}$$

The corresponding likelihood ratio test is

$$\Lambda(\mathcal{S}_m) \lessgtr^{H_1}_{H_0} \eta \tag{63}$$

where H_0 and H_1 corresponds to hypotheses that the measurement $\mathcal{S}_m$ is associated to the point $\mathcal{P}_{i_m}$ and to the line $\mathcal{L}_{j_m}$, respectively. The threshold is selected as $\eta < 1$, since (52) is two dimensional and (56) is one dimensional. More theory about likelihood test is given in e.g. [39].

No two measurements may originate from the same point source and no two sources may give rise to the same measurements. However, one line source may give rise to multiple measurements. This means that if measurement $\mathcal{S}_m$ is associated to point $\mathcal{P}_i$, then the values in the m^{th} row of the two matrices as well as the i^{th} column of the point likelihood matrix must be set to zero to exclude the measurement and the point from further association. However, if $\mathcal{S}_m$ is associated to line $\mathcal{L}_j$, then only the values in the m^{th} rows of the two matrices are set to zero because the line $\mathcal{L}_j$ can still be associated to other measurements. The procedure is repeated until all measurements with non-zero likelihood have been associated to either a point or a line. A new point is initiated if the observations could not be associated to an existing state. This is true when a measurement is not in the gate of a non-associated point or a line.

To summarize this section we state the main steps we used to associate measurements to tracked points or lines in Algorithm 4.1.

Algorithm 4.1 (Measurement Association)

---

1. **Likelihood matrices** *of all combinations of measurements and tracked point or lines are derived according to* (52) *and* (56)*, respectively. Form the likelihood matrices*

$$\Gamma_{\mathcal{P}}(\mathtt{m},\mathtt{i}) = \ell_{\mathcal{S}_m\mathcal{P}_i}$$

and

$$\Gamma_{\mathcal{L}}(\mathtt{m},\mathtt{j}) = \ell_{\mathcal{S}_m\mathcal{L}_j}$$

2. **Find the Maximum Likelihood** *value of* $\Gamma_{\mathcal{P}}$ *and the corresponding maximum value of* $\Gamma_{\mathcal{L}}$ *for the same measurement*

```
[max_lh,points] = max(Γ_P);
[dummy,meas_index] = max(max_lh);
point_index = points(meas_index);
[dummy,line_index] = max(Γ_L(line_index,:));
```

3. **Hypotheses test** *is given in* (63) *and the likelihood ratio* $\Lambda(\mathcal{S}_m)$ *is defined in* (62)

```
if Γ_P(meas_index,point_index)/Γ_L(meas_index,line_index) > η
    % associate with point
    Γ_L(:,point_index) = 0;
else
    % associate with line
end
Γ_P(meas_index,:)  = 0;
Γ_L(meas_index,:)  = 0;
```

4. **Repeat** *from step 2 until all measurements with non-zero likelihood have been associated to a point or a line. New points are initiated from the remaining measurements.*

In this section we have derived a method for tracking stationary objects as extended objects using radar measurements. Typically, radar echoes stem from delineators or guardrails, which are tracked as points or lines, respectively, in a standard Kalman filter framework. A major part of the present approach is the data association and gating problem. The results from both highways and rural roads in Sweden are not perfect, but surprisingly good at times, and of course much more informative than just using raw measurements.

4.3 Handling Tracks

A line is initiated from tracked points under the assumption that a number of points form a line parallel to the road. In this section we will discuss track handling matters such as initiating and removing tracks.

4.3.1 Initiating Lines

All points $\mathcal{P}_i$ are transformed into the ego vehicles coordinate frame since the road's geometry is given in this frame. The road geometry is described by the polynomial given in (1). We consider hypothetical lines passing through each point $\mathcal{P}_k$ parallel to the road. For each such line, the corresponding lateral distance $l_{\mathcal{P}_k}$ is given by

$$l_{\mathcal{P}_k} = \hat{y}^E_{\mathcal{P}_k O^E} - \delta_r \hat{x}^E_{\mathcal{P}_k O^E} - \frac{c_0}{2}\left(\hat{x}^E_{\mathcal{P}_k O^E}\right)^2. \tag{67}$$

The likelihood $\ell_{\mathcal{P}_i \mathcal{P}_k}$ that a point $\mathcal{P}_i$ is on the line of point $\mathcal{P}_k$ is then given by

$$\ell_{\mathcal{P}_i \mathcal{P}_k} = \begin{cases} \mathcal{N}\left(\epsilon_{\mathcal{P}_i \mathcal{P}_k}; 0, \mathbf{P}^E_{\mathcal{P}_k,(2,2)}\right), & \text{if } \begin{bmatrix} \hat{x}^E_{\mathcal{P}_i O^E} \\ \hat{y}^E_{\mathcal{P}_i O^E} \end{bmatrix} \in \mathcal{G}_{\mathcal{P}_k} \\ 0, & \text{otherwise,} \end{cases} \tag{68}$$

where the lateral distance between the point $\mathcal{P}_i$ and the proposed new line of point $\mathcal{P}_k$ is given by

$$\epsilon_{ik} = \hat{y}^E_{\mathcal{P}_i O^E} - \hat{y}^E_{\mathcal{P}_k O^E}, \tag{69}$$

where

$$\hat{y}^E_{\mathcal{P}_k O^E} = l_{\mathcal{P}_k} + \delta_r \hat{x}^E_{\mathcal{P}_i O^E} + \frac{c_0}{2}\left(\hat{x}^E_{\mathcal{P}_i O^E}\right)^2, \tag{70}$$

and the state covariance in the ego vehicles coordinate frame is given by

$$\mathbf{P}^E_{\mathcal{P}_k} = \left(R^{EW}\right)^T \mathbf{P}_{\mathcal{P}_k} R^{EW}. \tag{71}$$

The notation $\mathbf{P}^E_{\mathcal{P}_k,(2,2)}$ refers to the lower-right element, i.e., the variance in the diagonal corresponding to y^E. The gate $\mathcal{G}_{\mathcal{P}_k}$ is defined as

$$\mathcal{G}_{\mathcal{P}_k} \triangleq \Bigg\{ \begin{bmatrix} x \\ y \end{bmatrix} \Bigg| \left[y - \hat{y}^E_{\mathcal{P}_k O^E}\right]^T \left(P^E_{\mathcal{P}_k,(2,2)}\right)^{-1} \left[y - \hat{y}^E_{\mathcal{P}_k O^E}\right] \leq \delta_{\mathcal{L}}, \\ -\delta_s < x - \hat{x}^E_{\mathcal{P}_k O^E} < \delta_e \Bigg\}. \tag{72}$$

From all combinations of likelihoods we form a symmetric matrix $\Gamma_{\mathcal{I}}$. The columns of $\Gamma_{\mathcal{I}}$ are summed and the maximum value corresponding to column k_m is chosen. If this column contains more than a certain number κ of non-zero rows, corresponding to points

$$\mathcal{P}_l = \{\mathcal{P} \mid \Gamma_{\mathcal{I}}(:, k_m) \neq 0\} \tag{73}$$

within the gate of $\mathcal{P}_{k_m}$, a line is formed from the points $\mathcal{P}_l$. The new line's states a_0, a_1 and a_2 are estimated by solving a least square problem using the points $\mathcal{P}_l$. The states s and e are the minimum and maximum x-coordinate value of the points, respectively. All elements in column k_m and rows i_m are set to zero and the procedure is repeated until no column contains more than κ non-zero elements.

We state the main steps used to convert points into new lines in Algorithm 4.2.

Algorithm 4.2 (Initiate Lines from Points)

1. **Transform points** *from world to ego vehicle coordinate frame*

   $$\texttt{point\_ego(:,i)} = R^{EW}\hat{\mathbf{x}}_{\mathcal{P}_i} - d^W_{O^E O^W}$$

   for $i = 1, \ldots, N_{\mathcal{P}}$ *where* $N_{\mathcal{P}}$ *is the number of tracked points.*

2. **Form hypothetical lines** *passing through each point* $\mathcal{P}_k$ *parallel to the road. For each such line, the corresponding lateral distance* $l_{\mathcal{P}_k}$ *is given by* (67).

3. **Test if point is in gate** *and store the value in a matrix*

   ```
   if [x̂^E_{P_i O^E}; ŷ^E_{P_i O^E}] ∈ G_{P_k}
       in_gate(i,k)=1;
   end
   ```

   and sum the columns

   ```
   num_in_gate = sum(in_gate);
   ```

4. **Derive the likelihood matrix** *for all combinations of hypothetical lines and points according to* (68)

   $$\Gamma_{\mathcal{I}}\texttt{(i,k)} = \ell_{\mathcal{P}_i\mathcal{P}_k}$$

   and sum the columns

   $$\texttt{lh\_sum} = \texttt{sum}(\Gamma_{\mathcal{I}})\texttt{;}$$

5. **Find the maximum likelihood** *value of* `lh_sum`

   ```
   [max_lh,index] = max(lh_sum);
   ```

6. **Test** *if more than* κ *points of column* `index` *are within the gate* (72) *form a new line from these points, according to* (73)

```
if num_in_gate(index) >= κ
    P_l = find(in_gate(:,index)==1);
    [a0,a1,a2] = [ones(num_in_gate(index),1),...
        point_ego(1,P_l)',point_ego(1,P_l)'.^2]\point_ego(2,P_l)';
    s = min(point_ego(1,P_l));
    e = max(point_ego(1,P_l));
    Γ_I(l,:)  = zeros(num_in_gate(index),N_P);
end
Γ_I(:,index) = zeros(N_P,1);
```

7. **Repeat** *from step 5 until no element in* `num_in_gate` *is greater than* κ.

4.3.2 Remove Lines or Points

For each state we introduce a counter. The counter is increased if the state is updated with new measurements and decreased if it was not updated during one iteration. A state is removed if the counter is zero.

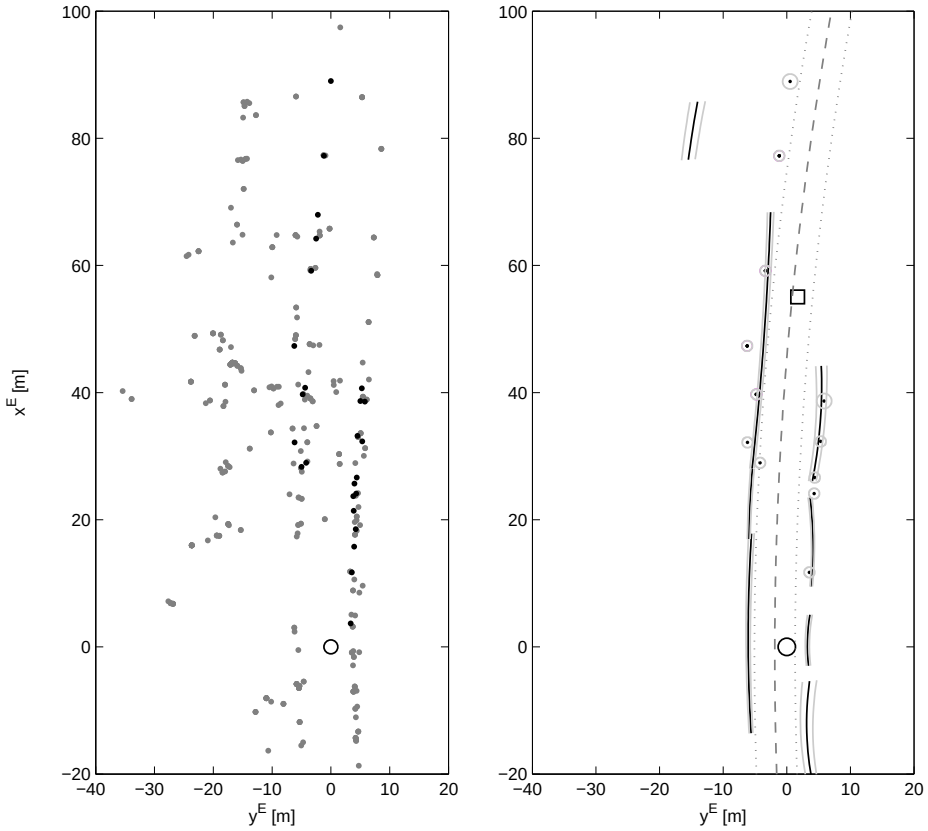
4.4 Experiments and Results

Let us start by showing the information given by an ordinary automotive ACC radar, for the traffic situation shown in Figure 9a. The ego vehicle, indicated by a circle, is situated at the $(0, 0)$-position in Figure 9b, and the black dots are the radar reflections, or stationary observations, at one time sample. The gray dots are former radar reflections, obtained at earlier time samples. Figure 9c shows the estimated points and lines for the same scenario. The mean values of the states are indicated by solid black lines or points. Furthermore, the state variance, by means of the 1σ confidence interval, is illustrated by gray lines or ellipses, respectively. In [25] the authors presented a new approach to estimate the road curvature (1), which we show here as gray dashed lines. We also show the tracked vehicle in front of the ego vehicle illustrated by a square.

In Figure 10a we see a traffic scenario with a highway exit. The corresponding bird's eye view is shown in Figure 10b. The origin of the line's coordinate systems are illustrated with dots and a number which is repeated at each line. Line 1 indicates the guardrail to the right of the exit, line 2 is the guardrail starting at the exit sign. The gap between line 3 and line 5 is probably due to the dimple, where the radar signals are transmitted above the guard rail, hence not giving us any stationary observations in the desired region.

Our last example shows a situation from a rural road, see Figure 11a. The lines 5 and 6 are the guardrails of a bridge. Line 4 depicts a fence behind the bridge. From the camera view it is hard to recognize and also the radar has problems to track it, indeed the gray lines indicates a large uncertainty for this case.

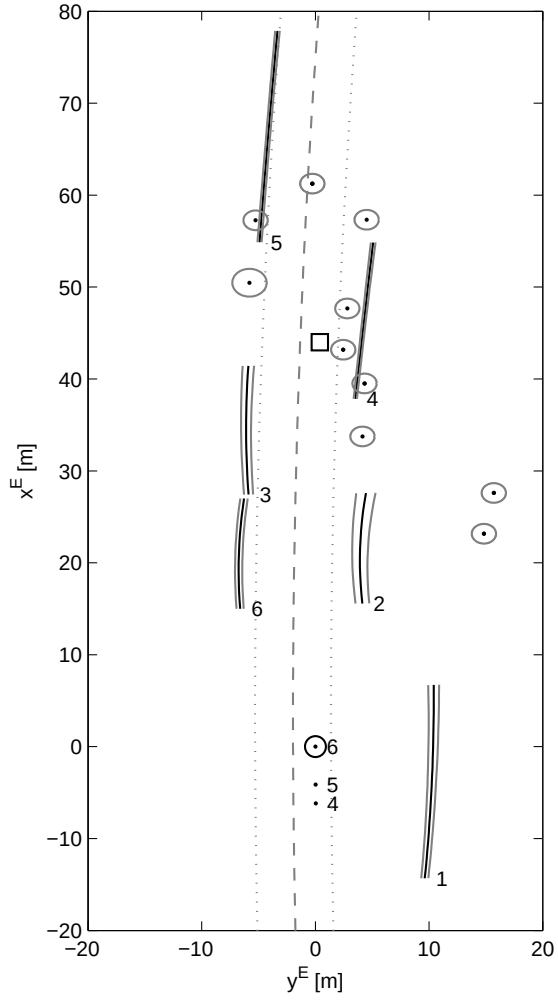
(a)



(b) (c)

Figure 9: A traffic situation is shown in Figure (a). Figure (b) shows the radar reflections, and Figure (c) the resulting tracked points and lines. The circle is the ego vehicle, the square is the tracked vehicle in front and the dashed gray lines illustrates the tracked road curvature.

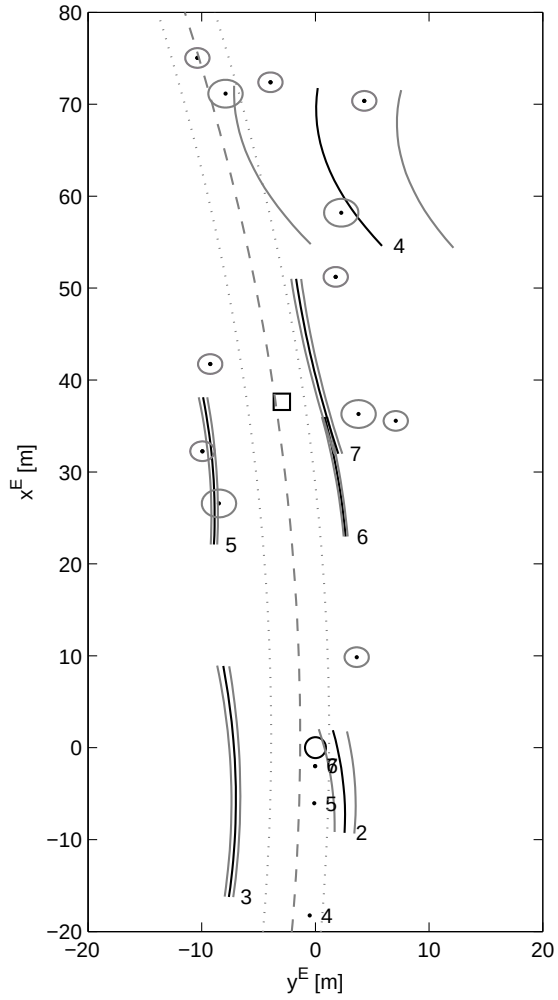
(a)



(b)

Figure 10: Highway exit with guardrails, the camera view is shown in Figure (a) and the bird's eye view with the estimated states in Figure (b).

(a)



(b)

Figure 11: A traffic scenario from a rural road, with guardrails on both sides of a bridge is shown. Note that the fence behind the bridge in Figure (a) is illustrated by line 4 in Figure (b), observe the large uncertainty.

4.5 Summary of Extended Object Tracking

We have derived a method for tracking stationary objects as extended objects using radar measurements. Typically, radar echoes stem from delineators or guardrails, which are tracked as points or lines, respectively, in a standard Kalman filter framework. The data association and gating problem is an important part of the present approach.

The main steps introduced in this section are repeated and summarized in Algorithm 4.3.

Algorithm 4.3 (Extended Object Tracking)

1. **Initialize** *all track counters and define the likelihood ratio test threshold* η.
2. **Transform measurements** *from polar coordinates (range* d *and bearing* $\psi_{\mathcal{S}_m O^E}$*) into Cartesian coordinates*

$$d^E_{\mathcal{S}_m O^E} = d_m \begin{bmatrix} \cos \psi_{\mathcal{S}_m E} \\ \sin \psi_{\mathcal{S}_m O^E} \end{bmatrix}$$
$$d^W_{\mathcal{S}_m O^W} = R^{WE} d^E_{\mathcal{S}_m O^E} + d^W_{O^E O^W}$$

   for $m = 1, \ldots, N_{\mathcal{S}}$ *where* $N_{\mathcal{S}}$ *is the number of current received measurements.*
3. **Measurement association** *as described in Algorithm 4.1.*
4. **Time update** *for every tracked point and line using the motion model in (46) and a Kalman filter.*
5. **Measurement update** *for every tracked point and line using a Kalman filter with (49) and (50), respectively. Increase track counter.*
6. **Initiate lines** *is described in Algorithm 4.2.*
7. **Remove tracks** *if the track counter becomes zero. The counter is decreased if a track was not updated during one iteration. Remove points which were converted into lines in step 6.*
8. **Merge lines** *overlapping lines with similar state values.*
9. **Repeat** *from step 2.*

5 Conclusions and Future Work

In this contribution we have derived three different methods for estimating the free space in front of a moving vehicle, making use of radar measurements originating from stationary objects along the road side. There is no need to introduce any new sensors, since the radar sensor is already present in modern premium cars. It is just a matter of making better use of the sensor information that is already present.

The presented methods do not depend on the fact that a radar sensor is used, implying that it is straightforward to add more sensor information from additional sensors. In other words, the approach introduced here fits well within a future sensor fusion framework, where additional sensors, such as cameras and additional radars, are incorporated.

The approaches have been evaluated on real data from both highways and rural road in Sweden. The results are encouraging and surprisingly good at times. They are of course not always perfect, but much more informative than just using the raw measurements. The problems typically occur when there are too few measurements or if the measurements stem from other objects than the road side objects.

5.1 Comparison

Three approaches to solve the same problem have been described in this paper. In this section we compare and summarize their properties, as well as their advantages and disadvantages.

The results of the presented methods are better than expected, given the fact that we are only using measurements delivered by standard automotive sensor. The main drawback of the presented methods is that the result can be unstable or erroneous if there are too few measurement points or if the measurements stems from other objects than the guardrail.

The representation form of the OGM is a square matrix, of size N_{ogm}, with the log odds of each grid cell. Since most of the environment is unknown many of the matrix' elements are equal to the initial log odds. In our example, we used a 401×401 matrix implying that the environment is described by 160801 parameters. The number of parameters used for the QP method is 8 and 12 for the linear and nonlinear model, respectively. The star end endpoint of valid segments can be limited by the user. Our experience is that this vector is not longer than 100. A line modeling the extended objects are represented by 5 parameters and one coordinate frame which is defined by its position and heading, i.e., 3 parameters. We have observed at maximum 20 lines, adding up to 160 parameters. However, we suggest the user to limit the number of lines to 10, adding up to 80 parameters.

The computational time does certainly depend on the hardware on which the algorithm is implemented, but it is worth comparing the proposed algorithms. The average computation times over a sequence of 1796 samples for the presented methods are given in Table 3. Note that the times given in this table includes the complete algorithms including initialization and coordinate frame transformations. The times given in Table 2 only compares the optimization algorithms. All of the algorithms can be made more efficient by fine tuning the code. However, the potential of the extended object tracking is assumed to be highest, since time implicitly depends on the number of tracked objects, which can be reduced by merging tracks and associating measurements to fewer tracks.

The flexibility of the OGM and the extended object tracking must be said to be higher. The OGM is not tied to any form of the road boarder or the stationary objects. The extended objects can be modeled in various types of shapes. The QP method is least flexible in that it only models the left and right border lines.

Table 3: Average computational time for one sample.

Method	Time [ms]
Occupancy Grid Mapping, Section 2	14.9
Linear Predictor, Section 3	109.5
Nonlinear Predictor, Section 3	137.2
Extended Object Tracking, Section 4	28.6

5.2 Future Work

Currently there is a lot of activity within the computer vision community to be able to handle non-planar road models, making use of parametric models similar to the ones used in this paper. A very interesting avenue for future work is to combine the idea presented in this paper with information from a camera about the height differences on the road side within a sensor fusion framework. This would probably improve the estimates, especially in situations when there are too few radar measurements available.

Acknowledgement

The authors would like to thank Dr. Andreas Eidehall at Volvo Car Corporation for providing data and for initial discussions on the topic. The idea of using the arctan-function in the predictor was brought to our attention by Professor Anders Hansson. Furthermore, we would like to thank the SEnsor Fusion for Safety (SEFS) project within the Intelligent Vehicle Safety Systems (IVSS) program and the strategic research center MOVIII, funded by the Swedish Foundation for Strategic Research (SSF) for financial support.

References

[1] D. Angelova and L. Mihaylova. Extended object tracking using Monte Carlo methods. *IEEE Transactions on Signal Processing*, 56(2):825–832, February 2008.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Mathematics in science and engineering. Academic Press, Orlando, FL, USA, 1988.

[3] Å Björck. *Numerical methods for least squares problems*. SIAM, Philadelphia, PA, USA, 1996.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] M. Buehler, K. Iagnemma, and S. Singh, editors. Special Issue on the 2007 DARPA Urban Challenge, Part I, volume 25 (8). *Journal of Field Robotics*, 2008.

[6] M. Buehler, K. Iagnemma, and S. Singh, editors. Special Issue on the 2007 DARPA Urban Challenge, Part II, volume 25 (9). *Journal of Field Robotics*, 2008.

[7] M. Buehler, K. Iagnemma, and S. Singh, editors. Special Issue on the 2007 DARPA Urban Challenge, Part III, volume 25 (10). *Journal of Field Robotics*, 2008.

[8] E. D. Dickmanns. *Dynamic Vision for Perception and Control of Motion*. Springer, 2007.

[9] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.

[10] B. Fardi, U. Scheunert, H. Cramer, and G. Wanielik. Multi-modal detection and parameter-based tracking of road borders with a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 95–99, June 2003.

[11] T. Fukae, N. Tamiya, and H. Mandai. Lateral distance measurement using optical spread spectrum radar. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1–6, Tokyo, Japan, September 1996.

[12] A. Gern, U. Franke, and P. Levi. Advanced lane recognition - fusing vision and radar. In *Proceedings of the IEEE Intelligent Vehicles Symposimum*, pages 45–51, Dearborn, MI, USA, October 2000.

[13] A. Gern, U. Franke, and P. Levi. Robust vehicle tracking fusing radar and vision. In *Proceedings of the international conference of multisensor fusion and integration for intelligent systems*, pages 323–328, Baden-Baden, Germany, August 2001.

[14] K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. In *IEE Proceedings of Radar, Sonar and Navigation*, volume 152, pages 364–371, October 2005.

[15] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Inertia-controlling methods for general quadratic programming. *SIAM Review*, 33(1):1–36, March 1991.

[16] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33, 1983.

[17] International Organization for Standardization ISO. *Passenger cars – Test track for a severe lane-change manoeuvre – Part 1: Double lane-change*. ISO 3888-1:1999. Geneva, Switzerland, 1999.

[18] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge. An algorithm for detecting roads and obstacles in radar images. *IEEE Transactions on Vehicular Technology*, 50(1):170–182, January 2001.

[19] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

[20] A. Kirchner and C. Ameling. Integrated obstacle and road tracking using a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 675–681, Dearborn, MI, USA, October 2000.

[21] A. Kirchner and T. Heinrich. Model based detection of road boundaries with a laser scanner. In *Proceedings of the IEEE International Conference on Intelligent Vehicles*, pages 93–98, Stuttgart, Germany, 1998.

[22] J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.

[23] K.R.S. Kodagoda, W.S. Wijesoma, and A.P. Balasuriya. CuTE: Curb Tracking and Estimation. *IEEE Transactions on Control Systems Technology*, 14(5):951–957, Sept. 2006.

[24] S. Lakshmanan, K. Kaliyaperumal, and K. Kluge. Lexluther: an algorithm for detecting roads and obstacles in radar images. In *Proceedings of the IEEE Conference on Intelligent Transportation System*, pages 415–420, Boston, MA, USA, November 1997.

[25] C. Lundquist and T. B. Schön. Road geometry estimation and vehicle tracking using a single track model. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 144–149, Eindhoven, The Netherlands, June 2008.

[26] C. Lundquist and T. B. Schön. Road geometry estimation and vehicle tracking using a single track model. Technical Report LiTH-ISY-R-2844, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, March 2008.

[27] C. Lundquist and T. B. Schön. Estimation of the free space in front of a moving vehicle. In *Proceedings of the SAE World Congress*, Detroit, MI, USA, April 2009. Accepted for publication.

[28] B. Ma, S. Lakshmanan, and A.O. Hero. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):135–147, September 2000.

[29] R.J. Mayhan and R.A. Bishel. A two-frequency radar for vehicle automatic lateral control. *IEEE Transactions on Vehicular Technology*, 31(1):32–39, 1982.

[30] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: Servey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, March 2006.

[31] M. Nikolova and A Hero. Segmentation of a road from a vehicle-mounted radar and accuracy of the estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 284–289, Dearborn, MI, USA, October 2000.

[32] A. Polychronopoulos, A. Amditis, N. Floudas, and H. Lind. Integrated object and road border tracking using 77 GHz automotive radars. In *IEE Proceedings of Radar, Sonar and Navigation*, volume 151, pages 375–381, Dec. 2004.

[33] M.J.D. Powell. On the Quadratic Programming Algorithm of Goldfrab and Idnani. *Mathematical Programming Study*, 25(1):46–61, 1985.

[34] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):84–90, 1979.

[35] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle filters for tracking applications*. Artech House, London, UK, 2004.

[36] J. Sparbert, K. Dietmayer, and D. Streller. Lane detection and street type classification using laser range images. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 454–459, Oakland, CA, USA, August 2001.

[37] N. Tamiya, H. Mandai, and T. Fukae. Optical spread spectrum radar for lateral detection in vehicles. In *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications Proceedings*, pages 195–198, Mainz, Germany, September 1996.

[38] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, Cambridge, MA, USA, 2005.

[39] H. van Trees. *Detection, Estimation, and Modulation Theory*. John Wiley & Sons, New York, USA, 1968.

[40] J. Vermaak, N. Ikoma, and S. J. Godsill. Sequential Monte Carlo framework for extended object tracking. *IEEE Proc.-Radar Sonar Navig.*, 152(5):353–363, October 2005.

[41] Vägverket, Swedish Road Administration, Borlänge, Sweden. *Vägar och gators utformning – Landsbygd - Vägrum*, 2004. 2004:80.

[42] Y. Wang, L. Bai, and M. Fairhurst. Robust road modeling and tracking using condensation. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):570–579, Dec. 2008.

[43] A. Wedel, U. Franke, H. Badino, and D. Cremers. B-spline modeling of road surfaces for freespace estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 828–833, Eindhoven, The Netherlands, June 2008.

[44] W.S. Wijesoma, K.R.S. Kodagoda, and A.P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *IEEE Transactions on Robotics and Automation*, 20(3):456–464, June 2004.

[45] Z. Zomotor and U. Franke. Sensor fusion for improved vision based lane recognition and object tracking with range-finders. In *Proceedings of IEEE Conference on Intelligent Transportation System*, pages 595–600, Boston, MA, USA, November 1997.

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2009-04-06

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ _____

Rapporttyp
Report category

☐ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☒ Övrig rapport
☐ _____

ISBN
—

ISRN
—

Serietitel och serienummer
Title of series, numbering

ISSN
1400-3902

LiTH-ISY-R-2892

URL för elektronisk version
http://www.control.isy.liu.se

Titel
Title

Estimation of the Free Space in Front of a Moving Vehicle

Författare
Author

Christian Lundquist, Thomas B. Schön, Umut Orguner

Sammanfattning
Abstract

There are more and more systems emerging making use of measurements from a forward looking radar and a forward looking camera. It is by now well known how to exploit this data in order to compute estimates of the road geometry, tracking leading vehicles, etc. However, there is valuable information present in the radar concerning stationary targets, that is typically not used. The present work shows how radar measurements of stationary targets can be used to provide a reliable estimate of the drivable space in front of a moving vehicle.

In the present paper three conceptually different methods to estimate stationary objects or road borders are presented and compared. The first method considered is occupancy grid mapping, which discretizes the map surrounding the ego vehicle and the probability of occupancy is estimated for each grid cell. The second method applies a constrained quadratic program in order to estimate the road borders. The problem is stated as a constrained curve fitting problem. The third method associates the radar measurements to extended stationary objects and tracks them as extended targets.

The required sensors, besides the standard proprioceptive sensors of a modern car, are a forward looking long range radar and a forward looking camera. Hence, there is no need to introduce any new sensors, it is just a matter of making better use of the sensor information that is already present in a modern car. The approach has been evaluated and tested on real data from highways and rural roads in Sweden and the results are very promising.

Nyckelord
Keywords

SLAM, Occupancy Grid Map, Guard Rail Detection, Extended Object Tracking