Camera based detection and classification of soft shoulders, curbs and guardrails

Alexander Seibert  Michael Hänel  Andreas Tewes  Raúl Rojas

Abstract—This paper describes the extension of an existing commercial lane detection system for marked roads by the detection and classification of soft shoulders, curbs and guardrails. Available sensors are a front directed grayscale camera and the CAN-bus data. To accomplish our goals we use two complementing methods. The first method is meant for localizing the border using a texture based area classification. We use Local Binary Patterns for their quality as well as their ability for being realized on an embedded system and a neural network as classifier. The second method is a kind of structure from motion for identification of raised structures like guardrails. The algorithm tracks Harris features using the Lucas and Kanade tracker and extracts 3D information out of it. The whole system works in real time and achieves an availability of about 80\% with a detection rate of 94\% on a test database, which includes roads under almost all weather conditions throughout a number of countries.

I. INTRODUCTION

There is a multitude of publications ([1], [2], [3], [4]) dealing with video based detection of marked roads. Though this is already a big achievement, as a next step autonomous vehicles and driving assistance systems have to become more universal, by being able to identify unmarked roads as well. This paper extends the architecture of an existing commercial Lane Departure Warning(LDW), which is based on the method of Behringer[5]. In short, the algorithm works as follows: pairs of ascending and descending edges are expected to be part of a marking. Near pairs are merged to segments, and segments are transformed to world coordinates (fig. 1) and grouped to tracks. Selected points, so called hypotheses, of the tracks are put into an extended Kalman filter, whose output is a smoothed clothoid model of the current lane pavement. Both sides of the lane are computed completely independently.

![Fig. 1. Used coordinate systems a) world coordinates (blue), camera coordinates (yellow) b) image coordinates](image1)

Our goal was the integration of a hypothesis generation for non-marked roads in that infrastructure. The algorithm is expected to be active only if there are no markings detected and should create its own road model.

In general it is very difficult to identify the exact position of the end of the lane due to the diversity of possible road borders (fig. 2). We assumed that the road has a different kind of structure, in comparison with the adjacent area. Due to this assumption we used texture based features, Local Binary Patterns[6] in particular, for a segmentation of the road. The benefits of that method against some edge based algorithms are the independence from clearly defined border and the ability to distinguish between soft shoulders and curbs.

To complete the identification of the lane it is also interesting if there is a guardrail beside it. An algorithm uses Harris features [7] and Lukas and Kanade image tracking [8] to get information about the structure from the camera motion. That is used for the detection of guardrails and walls.

II. RELATED WORKS

Vision based lane detection is the topic of numerous publications, but only a few deal with non-marked roads. Most of them are designed for autonomous vehicles [9], [10], equipped with a huge number of different sensor systems and expensive hardware. In the literature there are four different approaches: methods based on edge detection, on texture filter, on color segmentation and on 3D segmentation.

Many of the commercial driving assistance systems are still working with gray scale, so color segmentation as described in [11], [12] is not applicable at present. Most of the methods for recognition of marked roads are based on edge detection, so some authors use a similar approach for non-marked roads. The diverse properties of the edge detectors, noise reduction and the plausible road model [13], [14], [15], [16], [17], [18] form the crucial point of these methods.

The approaches of texture based applications are manifold. They reach from the usage of Gabor wavelets for determining

![Fig. 2. Examples of not clearly defined lane borders](image2)
the vanishing point[19], via the identification of the main
direction, the interpretation of the image as a graph [9] to
the segmentation according to road and environment [16],
[20].
3d information highly increases the quality of driving assis-
tance systems. Some authors use stereo vision or additional
sensors like lidar or radar to detect curbs[21], [22], [23],
[24], [25], or the whole road [26].

III. BORDER RECOGNITION

A. Border detection

Our detection algorithm is based on the Local Binary
Patterns (LBPs) [6]. This is a texture filter, used successfully
in face detection and some segmentation algorithms. The
Local Binary patterns are very simple and therefore fast
computable texture descriptors, based on the relation between
the gray scale values of a center pixel and its neighbors
(Fig. 3). The values of the LBP features in a defined
area are put into a histogram, which in return was used
as input vector for a classifier. Various radius modes [27]
and different preselections [28] were analyzed in several
studies. Experiments with different configurations showed
that the basic approach with 8 neighbors and radius 1 (fig.
3) performs best in our case. For training and testing we could refer to a high range of
videos, which have been recorded under almost all weather
conditions - in many European countries, as well as in
the US. Additionally, these videos have been labeled with
ground truth. One subset was used exclusively for training.
An algorithm randomly selects rectangular regions in each
frame and saves their LBP histograms and belongings to
one of the following classes (fig. 4): road, non-road, line,
interrupted line, curb and soft shoulders. At last we collected
a database of about 300k histograms.

Experiments with different rectangle sizes and classifiers
showed best performance with size 15 x 25 and a Multi
Layer Perceptron neural network (MLP)[29]. The training
used only four of the classes: road, non-road, curb and soft
shoulders. That is because the basic LDW system already
detects markings and less classes means better performance
of the MLP.

Once we were able to classify any area in the image, we
should find some border-points for matching our road model
on them. The obvious approach was a complete segmentation
of the image (Fig. 5), which worked well in recognition
but was too slow for real time applications. For runtime
improvement the following search algorithm was introduced:
we assign an area in front of the vehicle, expected to be part
of the road. The idea is to find some starting points \( q_i \), from
which we know that they are on the road and walk along the
same image row through the position of the border.
Let the image point \( q_i \) be the first approximation for a
point \( q_i' \) which is positioned on the right lane border, on the
condition that \( q_i \) is part of the road and both points have the
same y-coordinate. For calculating \( q_i' \) we start at the position
of \( q_i \), increase its x-coordinate and classify the area around
the new point with LBPs until the output of the classifier
has a high probability of one of the non-road classes(non-
road, curb or soft shoulder). That is a first assumption for the
position of \( q_i' \). For the verification of the result we observe
some other areas around the point (Fig. 6) and then \( q_i' \) can be
treated as a hypothesis and put into the Kalman filter. The
calculation of a point on the left side works in an analog
fashion. The success of that algorithm depends on the right
choice of the starting points. The starting points \( q_i \) should
fulfill several conditions:

- \( q_i \) should be on the road
- \( q_i \) should be as close as possible to the border for less
  iterations
- the number of starting points should be high enough for
  the computation of a road model

\[ 150 \ 120 \ 170 \]
\[ 90 \ 150 \ 200 \]
\[ \text{bit} = 1001011 \Rightarrow \]
\[ b_{16} = 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + \]
\[ 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^0 + 1 \times 2^0 = 299 \]

Fig. 3. Calculating one LBP feature: The relationship between the center
pixel and its neighbors is saved in the bitmask and the bitmask is used as
histogram entry

Fig. 4. Randomly selected areas from labeled videos. LBP histograms in
the rectangles are calculated and saved in the training database.Blue is road,
red is non-road and the gray scales represent different types of border

Fig. 5. complete segmentation. Blue is road, red is non-road, cyan is curb,
green is soft shoulder and empty field means that the probability for each
class is to small
• the number of starting points should be small enough for being computed in real time

For a few points we have defined an area in front of the vehicle, assumed to be part of the road (figure 7). The first set of points are not sufficient for a solid road model, that is why we need more starting points. For each known starting point \( q_{i,s}, i \in \{1..n\}, s \in \{L,R\} \) we compute the corresponding border point \( q_{i,s}^\prime \) as already described. Using a RANSAC algorithm [30] on the border points we assign two best suitable lines, \( g_L \) on the left side and \( g_R \) on the right side. That helps calculating the next set of starting points \( q_{j,L} \) and \( q_{j,R} \):

\[
\begin{align*}
  a &= g_R(y_{q_{j,R}} - \Delta y) - g_L(y_{q_{j,L}} - \Delta y) \\
  \bar{q}_{j,R} &= \left( g_L(y_{q_{j,L}} - \Delta y) + (1 - \epsilon) * a \right) \\
  \bar{q}_{j,L} &= \left( g_L(y_{q_{j,L}} - \Delta y) + \epsilon * a \right) \\
  \end{align*}
\]

(1)

where \( g_{R/L}(y) = x, (x,y) \in g_{R/L} \) and \( a \) is the distance between both lines in the next image row of interest. Figure 8 clarifies the procedure. The factor \( \epsilon \) is a buffer for the position of the starting points due to possible lane curvature.

Based on the border points we set up a road model. The current work uses the already existing and established method of Behringer [5]. They use a clothoid in world coordinates to describe the course of the road. The border points found in each frame are fused by the prediction of a Kalman filter [31]. The system is available if the tracking is stable over a few seconds.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig6}
\caption{Searching starts at point \( q_i \) and corrects its \( x \)-coordinate until it reaches the border point \( q_i^\prime \).}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig7}
\caption{Close starting points can be defined, far points should be calculated by another method.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig8}
\caption{Extraction of starting point: a) one iteration for recognizing the next start points \( q_{i,L}^\prime \) and \( q_{i,R}^\prime \) b) realized algorithm: blue are the start points on the right side and pink the start points on the left side.}
\end{figure}

B. Border classification

For the classification we use two different methods. The results from the localization can also be used for classification of soft shoulders and curbs. Detecting guardrails, however, requires an alternative approach since guardrails have no unique texture throughout the countries. Due to this fact we decided to use a structure from motion approach.

1) Curbs and soft shoulders: For the classification of curbs and soft shoulders our method evaluates the already calculated LBP histograms, which are on the road border. For each border point \( p_i, i \in \{1..n\} \) the output vector of the MLP \( \nu_{p_i} \), containing the probabilities for each class, so that \( \nu_{p_i}(j), j \in \{1..4\} \) is the probability for \( p_i \) being on a road \( (j = 1) \), a non-road \( (j = 2) \), a soft shoulder \( (j = 3) \) or a curb \( (j = 4) \) area. The average probability vector \( \nu \) allows a quite robust identification of the border. We can assume that the recognition of markings is solved by other methods, so the challenge is to differentiate curb, soft shoulder and wall. As there is often lawn directly behind a curb, the output vector of the neural network gives the same or higher probability for the soft shoulder. Therefore we defined a heuristic recognition \( \xi \) as follows:

\[
\xi = \begin{cases} 
  \text{curb, if } \nu(4) > \theta_4 \\
  \text{soft shoulder, if } \nu(3) > \theta_3 \land \nu(4) \leq \theta_4 \\
  \text{unknown, else} 
\end{cases}
\]

(2)

where \( \theta_x \) is an experimentally estimated threshold. To make the method robust we used a hysteresis over time.

2) Guardrails: The most important difference between guardrails and other border types is the vertical alignment on the road plain. Therefore it is reasonable to use a 3D approach for their classification. One promising way to get 3D information from a monocular camera system is structure from motion, considering information about the ego-motion. In our first step we use some standard algorithms: Detecting Harris corner [7] and track them over a predefined distance \( d \) with the method of Lukas and Kanade [8]. The problem is that roads, marks, and also guardrails are very regular structures, so the tracker sometimes "slides" on them. That is why we need some assumptions for verification of the tracked points:

• points above the vanishing line are of no importance
• the camera moves forward only
• the camera view is not exactly horizontal to the road, but is tilted slightly forward.

Under that constraint a tracked point \( p \) is verified if \( y_0 > y_1 > y_{vp} \) and where \( y_0 \) is the \( y \) value of \( p \) in the previous frame and \( y_{vp} \) is the position of the vanishing line.

Since we have a calibrated camera system, we are able to use the following functions:

- transition function \( f_{WtoI} : (x_{im}, y_{im})^T \mapsto (x_w, y_w)^T \) image to car coordinates with defined \( z_w = 0 \)
- transition function \( f_{WtoI} : (x_{im}, y_{im})^T \mapsto (x_w, y_w, z_w)^T \) car to image coordinates
- function \( h : (t_0, t_1) \mapsto (\Delta x_w, \Delta y_w, 0)^T \) calculates passed distance between moment \( t_0 \) and \( t_1 \) in \( x \) and \( y \) direction in car coordinates
- exact camera position in car coordinates (\( p_c \))

The goal is to determine the position of the tracked points in the car coordinates. Let \( t_0 \) be the moment of detecting the feature and \( t_1 \) the moment when the vehicle has passed the distance \( s \) since \( t_0 \). The point \( p \) is the position of the tracked feature in world coordinates, \( p_c = (x_c, y_c, z_c) \) and \( p'_c = (x'_c, y'_c, z'_c) \) are the positions of the camera at \( t_0 \) and \( t_1 \), respectively. The 2D point \( b_0 \) is the detected feature in image coordinates at timestamp \( t_1 \) and \( b_1 \) is the same feature at \( t_1 \). The values of \( t_0, t_1, b_0, b_1, p_c \) are known quantities sufficient to compute \( p \) (fig. 9):

\[
P_c = p_c + g(t_0, t_1); \quad (3)
\]
\[
P_0 = f_{WtoI}(b_0) \quad (4)
\]
\[
P_1 = f_{WtoI}(b_1) + g(t_0, t_1) \quad (5)
\]
\[
g_0 = \{ \tilde{\alpha}_0 + u\tilde{r}_0 | u \in \mathbb{R} \}, \quad \text{where} \quad \tilde{\alpha}_0 = \tilde{p}_c - \tilde{p}_0 = \tilde{p}_0 - \tilde{p}_c \quad (6)
\]
\[
g_1 = \{ \tilde{\alpha}_1 + u\tilde{r}_1 | u \in \mathbb{R} \}, \quad \text{where} \quad \tilde{\alpha}_0 = \tilde{p}_c - \tilde{p}_1 = \tilde{p}_1 - \tilde{p}_c \quad (7)
\]

The intersection point between the lines \( g_0 \) and \( g_1 \) is the point \( p \). Because of the measuring error both lines rarely intersect each other, so we have to calculate the closest point.

Due to the egomotion error there is a deviation of \( 0.15m \) in the computation of \( z \), so we consider that structures higher than \( 0.3m \), measured higher than \( 0.45m \), and close to the road border are guardrails. So let \( P = \{ p = (x, y, z) | z > 0.5, p \text{ is close to the border} \} \) of detected 3D points, there is a guardrail if \( |P| > \Theta \), where \( \Theta \) is an experimentally estimated threshold. This threshold depends on the quality of the video and the number of solid tracked features in the area.

IV. EXPERIMENTAL EVALUATION

There is no standard database for testing non-marked Lane detection methods, similar to face or human detection. By means of an automatic evaluation system we performed a test run on a rated distance of approximately 200km in different countries, under diverse weather conditions and on different kinds of roads - city, rural or highway (fig 10). The availability and accuracy are shown in (table I). The quite low availability on the right side has various reasons: very bad roads, high reflections and shadows. More important is that there are very few false positives. That and the track-losses of less then 1 per km shows a solid system. The classification between soft shoulder and curb performs as shown in table II. The guardrails are in separate statistics (table III), because marks and guardrails can occur at the same time. We also analyzed the runtime on different computer systems (table IV).

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATA ABOUT THE EVALUATED ROADS, THE AVAILABILITY AND THE TRUE POSITIVE RATE</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>non marked part of the data</td>
</tr>
<tr>
<td>system available</td>
</tr>
<tr>
<td>true positives with accuracy 30cm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BORDER CLASSIFICATION</strong></td>
</tr>
<tr>
<td>true positive</td>
</tr>
<tr>
<td>guardrail</td>
</tr>
<tr>
<td>no guardrail</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLASSIFICATION OF GUARDRAILS</strong></td>
</tr>
<tr>
<td>PC</td>
</tr>
<tr>
<td>test pc 1</td>
</tr>
<tr>
<td>test pc 2</td>
</tr>
<tr>
<td>test pc 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE IV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PERFORMANCE OF THE WHOLE SYSTEM, INCLUDING THE BASIC LDW, ON DIFFERENT SYSTEMS</strong></td>
</tr>
</tbody>
</table>
V. FUTURE WORKS

There are a lot of possibilities to improve our system. Because of the signal noise of the camera in the dark and the sensitivity of the LBPs, the system works in daytime only. This problem could be solved by smoothing the image, for example, with a Gauss Kernel [32] or by using weighted LBPs [33].

It is possible to make an online learning of the LBP histograms, taking for example the area directly in front of the vehicle as ground truth. The fusion with other methods, e.g. methods based on edge detection, is expected to increase the precision even further.
REFERENCES


