

Map-based Lane and Obstacle-free Area Detection

T. Kowsari, S. S. Beauchemin and M. A. Bauer

Department of Computer Science, The University of Western Ontario, London, ON, N6A-5B7, Canada

Keywords: Lane Detection, Stereo Vision, Particle Filters, Lane Maps.

Abstract: With the emergence of intelligent Advanced Driving Assistance Systems (i-ADAS), the need for effective detection of vehicular surroundings is considered a necessity. The effectiveness of such systems directly depends on their performance in various environments such as rural and urban roads, and highways. Most of the current lane detection techniques are not suitable for urban roads with complex lane shapes and frequent occlusions. We propose a map-based lane detection approach which can robustly detect the lanes in urban and rural environments, and highways. We also present an algorithm for detecting obstacle-free areas in detected lanes based on the stereo depth maps of driving scenes. Experiments show that our approach reliably detects lanes and obstacle free areas within them, even in case of partially occluded or worn-off lane markers.

1 INTRODUCTION

Today, almost every new vehicle has some form of Advanced Driving Assistance System (ADAS). From adaptive cruise control, collision avoidance, and lane crossing warning systems to parking assistance, ADAS has made driving a safer and more enjoyable task. While a simple driving assistance system still requires a wealth of information on the state of the vehicle and its relationship to the immediate environment, intelligent ADAS requires even more, including information on the state of the driver. Furthermore, the relative position and speed of other vehicles (and obstacles) constitute essential informational elements in the determination of lane-based safe and driveable areas directly located in front of the vehicle. In this contribution, we present an innovative lane detection system which combines GPS information and a global lane map with a forward facing vehicular stereo system to achieve robust lane detection. In addition, the stereo depth map enables the detection of lane-based, obstacle-free areas.

Lane detection may appear trivial, at least in its basic setting. For instance, a relatively simple Hough transform-based algorithm can be used to detect the host lane for a short distance ahead without any tracking. This method proves effective in roughly 90% of the highway cases (Borkar et al., 2009). However, lane detection is considered a very challenging task when lanes other than the host one, obstacles of all kinds, and sharp turns are taken into account. The absence of lane markers (or worn-off ones), various lane shapes and sizes, occlusion, illumination

changes, and weather conditions are among the reasons why lane detection is not as simple as it seems.

A recent lane and road boundary detection survey (Hillel et al., 2012) explored a large body of research on lane detection, including methods using gradient-based feature detection (Samadzadegan et al., 2006; Nieto et al., 2008; Sawano and Okada, 2006), steerable filters (McCall and Trivedi, 2006), box filters (Huang et al., 2009; Wu et al., 2008), and learning-based lane pattern recognition (Cheng et al., 2006).

Lane models, such as straight lines (Kim, 2008; Pomerleau, 1995; Rasmussen and Korah, 2005), parabolic curves (Huang et al., 2009; McCall and Trivedi, 2006), semi-parametric formulations such as splines (Kim, 2008), or active contours (Sawano and Okada, 2006) are found in the literature. Different model-fitting methods have been adopted including RANSAC (Sawano and Okada, 2006), particle swarms (Zhou et al., 2005), energy-based optimization (Sawano and Okada, 2006), genetic algorithms (Samadzadegan et al., 2006), and more. Despite this vast body of research, there are problems which yet remain to be satisfactorily addressed:

- Lane markings cannot be detected with range finders or other types of sensing that do not provide visible spectrum images. Even when sensors are adapted to lane marking detection, external problems arise, such as adverse weather, weak illumination, and worn-off markings, among others. Only a few authors in the literature have used specialized sensors such as line sensors (Narita et al., 2003) or GPS (Jiang et al., 2010) to assist the detection process. In this contribution we

demonstrate how GPS and vehicle speed obtained from the internal network of the vehicle (CAN-bus) may be used in the design of a robust lane detection algorithm.

- Except in a few instances (Kim, 2008; Huang et al., 2009), in almost the entire lane detection literature, lane models are not taking splitting and merging lanes (such as left turn lanes or opening and closing lanes) into account. Models often consist of parallel lanes without any distortion or starts and end to them. We have used a very simple yet flexible way of representing lanes such that all types of lanes can be represented and detected in most situations.
- Current lane detection algorithms are usually designed and tested either on highways or rural roads where sharp changes in lane position and orientation are not often observed. Our approach was tested successfully in dense urban areas where sharp turns, vehicle clutters, lane marker coverage, buildings, or other urban artifacts distract conventional lane detection algorithms.
- Most times, the most important lane from the point of view of the detection process is the host lane. However, in some cases we are interested in being able to describe a more complex environment such as the sum of lines adjacent to the host one.

We first provide a map-based framework which uses the GPS, vehicular speed, and a pre-loaded digital lane map as inputs to the lane detection algorithm. We then present the lane feature detection mechanism together with a particle swarm based tracking algorithm which fits the map with the lanes in the images. Subsequently, we use a simple yet effective stereo depth-based obstacle detection by which we find the obstacle-free lane areas in front of the vehicle.

This contribution is organized as follows: Section 2 introduces the global lane map and lane modeling. Section 3 provides lane features and the Particle Swarm Optimization (PSO) algorithm, Section 4 describes the obstacle detection mechanism and the method to compute the obstacle-free lane areas, Section 5 presents the experimental results, and Section 6 offers a conclusion.

2 LANE MODEL

We present a global lane model for lane detection. While this type of model is not very common in the literature, we believe that it provides key advantages

to the development of robust lane detection mechanisms. Using a lane map containing all lane paths and vehicle location on that map (with GPS or other methods for localization) facilitates the lane detection process and results in a more robust approach to the problem. To form the required lane maps, we annotated lanes in images provided by Google Earth satellite imagery.

In most of the methods found in the existing literature, it is generally assumed that the lane markers on the ground plane are approximately parallel. However, in reality, lane markers do not conform to this assumption. Even on roads where there is no splitting or merging of lanes, there are frequent lane shape distortions. In addition, most methods are concerned with the detection of the host lane only. We propose that modeling multiple lanes can significantly contribute to the robustness of lane detection algorithms, as any detectable part of a lane assists in preserving stability, especially in the absence of other cues. In light of this, it is believed that a robust model should have the following properties:

- The model should address the *observed* shapes of lanes.
- In addition to the detection of the host lane, the model should be able to detect visible adjacent lanes.
- The model should include splitting and merging lanes (for instance, left turn lane parts in the center of the road at intersections or highway merging lanes)

The model contains a number of splines which model the entire map of the region of interest. Each spline is a lane marker and consists of points whose absolute positions on the map are their GPS latitude and longitude. In addition, these splines are binned into grid buckets representing non-overlapping contiguous regions each $500m^2$ in size. The sum of these buckets cover the entire lane map.

Each time the vehicle records data (it does so at 30Hz), a search for spline buckets that are most probably visible occurs, given the vehicle's position and orientation, and the front stereo system viewing angle. The lane marking splines from the selected buckets are subsequently sorted in space with respect to the perpendicular of the direction of the vehicle, which amounts to a sorting from left to right in terms of visibility from the point of view of the stereo system.

With t sorted lane marking splines hypothetically forming $t - 1$ lanes and two out-of-road areas, and the position (latitude and longitude) and orientation (obtained with the vector formed from the last two GPS coordinates) of the vehicle, the positions of the



Figure 1: Images from the map building application **a)** (left): Splitting lanes **b)** (right): Several neighboring lanes.

splines are converted into the reference frame of the front stereo system (with its origin at the optical center of the left camera) in meter units. Each lane $L_i \in \{L_0, \dots, L_{t-1}\}$ is composed of two lane marking splines.

In order to specify the modalities of splitting and merging lanes, the model requires the opening and closing distances of the lanes from the vehicle. To address this, at each time interval, we assign $t-1$ variables $\text{LaneCloses}(i)$ for the closing distance of each lane and another $t-1$ variables $\text{LaneOpens}(i)$ with the same size for the opening distance of each lane. The opening distances for the lanes which are already open are set to 0, while the closing distances for the lanes that are not yet closed are set to ∞ .

Since we require our model to detect obstacle-free areas in the lanes, we considered another $t+1$ variables $\text{LaneBlocks}(i)$ which contain either ∞ to signify not blocked or a distance in meters indicating that there is an obstacle in this lane at that distance.

2.1 Spline Lane Marker Model

We adopted the Catmull-Rom spline formalism for the lane-marking splines (Catmull and Rom, 1974) since it interpolates the control points. For each spline segment between control points P_i and P_{i+1} , the spline is obtained with control points P_{i-1} to P_{i+2} as (Watt and Watt, 1991):

$$S(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \quad (1)$$

where $S(t)$ is either the x or y element of the coordinates of the curve points, $t \in [0, \dots, 1]$ and

$$M = \frac{1}{2} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & \frac{1}{2} & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

2.2 Generating the Lane Map

Google Earth satellite images are used to build the lane maps. Satellite images adequately fit our purposes as lane markers are not occluded by vehicles or other urban structures. These images can also be addressed directly by longitude and latitude which is desirable since we use GPS coordinates to locate the vehicle on the map and extract hypothetically visible lanes from the stereo images. We created an application which uses Google static API to obtain and display bird's eye images of the region of interest at requested positions. (see Figure 1). This application also allows a user to draw and edit splines as lane markers. The user is also able to navigate through the map and follow the road while drawing lanes. The resulting data is saved as a set of lane-marking splines, each of them containing a set of control points. In our experiments, we extracted a path that was traveled by the experimental vehicle within the city of London, Ontario. This path consists of 94 lanes and lane segments, including right and left turn lanes.

3 MODEL FITTING USING A PARTICLE FILTER

With the knowledge of the position and orientation of the vehicle within the lane map, we proceed to fit our lane model onto the detected lane features in the left stereo image.

Since the GPS data frequency (1Hz) is significantly slower than that of the front stereo system (30Hz), the most recent speed data of the vehicle obtained from the CANBus is used to extrapolate the most recent available GPS data to coincide with the most recent image frame from the front stereo system. This can be thought as a form of synchronization of the GPS device and the front stereo system. In addition, the GPS data has a relatively large error (we observed a $\pm 5\text{m}$ error), and can be used only as a seed for the lane fitting process.

With the approximate position and orientation of the vehicle, the visible parts of the lane map in the image can be identified. The lane-marking splines are projected onto the stereo left image and an optimization algorithm attempts to find the best relative change in the position and orientation of the vehicle which best fits the projection with the lane features in the image. This optimization yields two parameters δX and $\delta \theta$ which correct the current vehicle position and orientation obtained from the GPS at each frame.

In order to project the lane markers onto the image we need to know the ground plane equation parame-

ters in the camera coordinate system. Even though the ground plane parameters are very stable, we noticed that including a correction parameter $\delta\lambda$ representing the difference between the ground plane and the x, z plane of the stereo camera system improves the accuracy of the projection process by compensating for the unexpected tilt variations due to vehicle suspension.

3.1 Ground Plane Estimation

The ground parameters needed for projecting the lanes on the image can be computed from the depth map obtained from the stereo system. With rectified stereo images, finding disparities and hence depth map merely consists of a 1-D search with a block matching algorithm (our implementation uses the stereo routines from Version 2.4 of OpenCV) Assuming that the ground plane equation is of the form

$$ax + by + cz = d \quad (2)$$

where $\vec{n} = (a, b, c)$ is the unit normal vector to the plane, we pose

$$d = \frac{1}{\sqrt{a^2 + b^2 + c^2}} \quad (3)$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = d \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} \quad (4)$$

With the coordinates of 3D points in the reference system of the left camera

$$(X_i, Y_i, Z_i) \quad (5)$$

we can write

$$\mathbf{Ax} = \mathbf{B} \quad (6)$$

and solve for \mathbf{x} in the least-squares sense as

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (7)$$

where

$$\mathbf{A} = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}$$

Often times the ground surface leads to inordinate amounts of outliers, due in part to a lack of texture from the pavement or other driveable surfaces. With the sensitivity of least-squares to outliers being known, we resort to the use of RANSAC in selecting the inliers and obtain a robust estimation of the ground plane coefficients, in the following way:

1. randomly select three points from the 3D points believed to be representative of the ground plane

2. compute the coefficients of the plane defined by the randomly selected points using (6)
3. count the points whose distance to the plane is less than a threshold ϵ
4. repeat these steps n times where n is sufficiently large¹
5. among the n fits choose the largest inlier set which respect to ϵ and compute the coefficients of the ground plane this time using least-squares as in (7)

The plane parameters are averaged over a short period of time in order to stabilize them further. The coefficients of the plane are recomputed at each new stereo frame arrival. However, in cases when the number of depth values is low (poor texture, etc.) or other vision modules indicate the presence of a near obstacle, the coefficients of the ground plane are not recomputed, the previous parameters are used instead.

Introducing the tilt parameter $\delta\lambda$, the ground plane equation becomes:

$$ax + by + (c + \delta\lambda)z = d \quad (8)$$

3.2 Likelihood Function

The estimation of the best fit parameters between projected lane-marking splines and the detected lane features in the left stereo image is performed by defining a likelihood function

$$\mathcal{L}(z|\mathbf{x}) \quad (9)$$

where z is a particular parameter fit, and $\mathbf{x} = (\delta x, \delta\theta, \delta\lambda)$. Estimating this likelihood function requires first the detection of lane boundary features from the stereo imagery. Image features must satisfy a number of constraints before they can be considered as lane boundary features, such as being located on the ground plane, featuring a lighter gray level than that of the ground plane, and be contained within two significant gradient values of a predefined width (which depends on the observed depth).

The algorithm to detect lane boundary features is formally described in 1 and uses the left camera stereo image I and its depth map I_d as inputs to produce a Gaussian smoothed lane boundary feature image F , such as that displayed in Figure 2b. Constants found in the algorithm are α and β , used for computing the width expectation of the lane markings L_{max} , factored by their distance from the vehicle. Constants NL and LD indicate the state of the lane edge search. NL represents the state in which no lanes are detected,

¹Choosing $n > 20$ does not significantly improve the number of inliers with respect to ϵ .

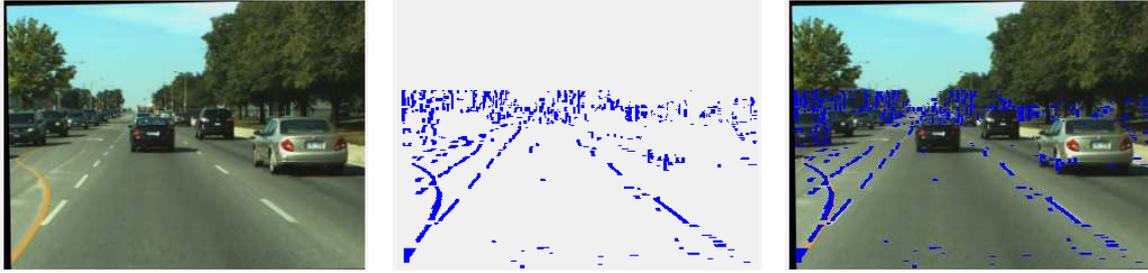


Figure 2: **a) (left):** Raw image **b) (center):** Low-level lane feature detection **c) (right):** Features depicted on the image.

while LD is its complement. Threshold τ_h represents the minimum gradient value required for a transition from NL to LD. Constant O_h is the minimum variation in height from the ground plane for a pixel to be considered part of an obstacle. O_h and τ_h depend on imagery and are experimentally determined.

Algorithm 1: Lane Feature Detection Algorithm.

```

 $G \leftarrow$  1D Gaussian row smoothing of  $I$  with  $\sigma = 0.5$ 
 $G \leftarrow$  horizontal gradient of  $G$  using 3-point central differences
Remove the values corresponding to obstacles from  $G$  using threshold  $O_h$ 
State  $\leftarrow$  NL
 $F$  initialized to 0
for all rows  $i$  in  $I$  starting from the image bottom do
     $L_{\max} \leftarrow \beta - i\alpha$ 
    Count  $\leftarrow$  0
    for all column  $j$  in  $I$  do
        if ( $G_{i,j} > \tau_h \wedge (\text{State} = \text{NL} \vee \text{Count} > L_{\max})$ ) then
            State  $\leftarrow$  LD
        end if
        if ( $\text{State} = \text{LD}$ )  $\wedge$  ( $G_{i,j} < -\tau_h$ ) then
            for  $k = j - \text{Count} \rightarrow j$  do
                 $F_{i,k} \leftarrow 1$ 
            end for
            State  $\leftarrow$  NL
            Count  $\leftarrow$  0
        end if
    end for
end for
 $F \leftarrow$  1D Gaussian row smoothing of  $I$  with  $\sigma = 0.5$ 

```

The likelihood function (9) may be estimated using the extracted lane marking features F and the sorted (from left to right) lane marking splines contained in the visible spline buckets. The lane-marking splines from the map are aligned with the direction of the vehicle by a rotation and then projected on the image plane so as to find a best fit with the detected lane marking features. Assuming that the Z axis of the 3D

reference frame of the front stereo system of the vehicle makes an angle θ with the Y axis of the 2D reference frame of the lane map, a spline point $\mathbf{Q} = (X, Y)$ in the coordinates of the lane map is rotated according to:

$$\begin{bmatrix} X_r \\ Z_r \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \cos(\theta) & -\sin(\theta) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (10)$$

With the ground plane equation, we estimate the tilt-corrected height coordinate in the reference frame of the stereo system as:

$$Y_r = \frac{d - aX_r - Z_r(c + \delta\lambda)}{b} \quad (11)$$

where $\mathbf{Q}_r = (X_r, Y_r, Z_r)$ is the 3D spline point expressed in the reference frame of the stereo system. The projection of \mathbf{Q}_r onto the stereo imaging plane is performed by applying the classical projection matrix P obtained for the calibration process of the stereo system:

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix} \quad (12)$$

where w is a scaling factor due to the use of homogeneous coordinates.

With the lane feature image F and the projected, visible lane-marking splines, the likelihood function becomes

$$\mathcal{L}(z|\mathbf{x}) = \sum_{(i,j) \in \mathbf{S}} F(i, j) \quad (13)$$

where \mathbf{S} is the set of all projected points of the lane-marking splines.

3.3 Particle Filtering

With the likelihood function, we need to estimate the parameters \mathbf{x} of the fit as:

$$x = \operatorname{argmax} \mathcal{L}(z|\mathbf{x})_{\mathbf{x}} \quad (14)$$

Solving this optimization problem is not easily achievable by regular hill-climbing methods due to

the non-concavity of the function. Since the search space is large, an exhaustive search is prohibitively expensive while the probability of finding the global maximum remains low (Talbi and Muntean, 1993).

A particle swarm method may be more appropriate. The particle swarm lane detection algorithm by Zhou (Zhou et al., 2005) is a single image frame method, which we adapt here as a particle filter working on a sequence of frames². Our approach consists of generating a set of uniformly distributed particles, each representing a set of possible values for parameters $\mathbf{x} = \delta x, \delta \theta, \delta \lambda$. The likelihood of each particle is estimated with (14).

At each iteration, each particle is replaced with a number of newly generated, Gaussian position-disturbed particles. The number of generated particles is proportional to the likelihood of the particle they replace. Their likelihood are estimated again with (14) and normalized. This ensures that the stronger particles generate more particles in their vicinity than the weaker ones. Particles with normalized likelihoods lower than a certain threshold are removed and, if the number of particles becomes less than a threshold, the process repeats.

These iterations eventually lead to groups of particles concentrated at the most likely answers in the search space and the particle with the maximum likelihood is chosen as the solution. In addition, keeping the particles over time makes the particle filter to act as a tracker for the lane detection mechanism.

4 OBSTACLE DETECTION

With a set of detected lanes represented by projected splines, the stereo depth map can be used to locate obstacles within each detected lane. The inputs to the obstacle-free detection algorithm are the stereo disparity map I_d , the classical projection and re-projection matrices P and D , the ground plane parameters a, b, c, d , and $\delta \lambda$, and the projected lane-marking splines. The output consists of the distance from the vehicle to first obstacle (if present) for each lane. The algorithm uses constant O_h as previously defined, and threshold O_t which is the minimum ratio of obstacle pixels to all pixels across a lane, for each row in the image.

The first stage of the algorithm consists of detect-

²PSO is a population-based stochastic optimization method first proposed by Eberhart and Kennedy (Kennedy and Eberhart, 1995).



Figure 3: **a) (top-left):** Color-coded stereo depth map **b) (top-right):** Accumulated projected obstacle points **c) (bottom):** Obstacle-free area detection.

ing pixels whose 3D positions computed as:

$$W \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = D \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix} \quad (15)$$

are not lying on the ground plane. The distance of the 3D point from the ground plane is obtained as:

$$\text{Dist} = aX + bY + (c + \delta \lambda)Z - d \quad (16)$$

The algorithm keeps an obstacle map O the size of the original image. The 3D coordinates of each pixel whose height from the ground plane qualifies it as an obstacle is projected onto the ground plane by setting its Y coordinate according to (11), and then projected onto the obstacle map O , using

$$w \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y_g \\ Z \\ 1 \end{bmatrix} \quad (17)$$

where the corresponding image location in O is incremented by one.

The last stage of the algorithm consists of scanning all rows of image O from the bottom. In each row, between the boundaries of each lane which is not yet blocked, the values of O at the positions across the lane are summed up and divided by the total number of pixels in that lane, forming a lane ratio γ . If this

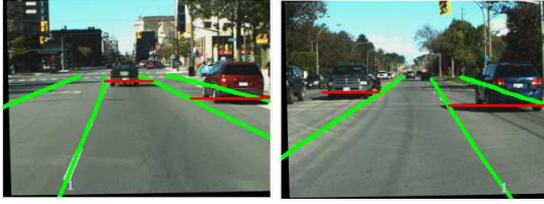


Figure 4: Examples of obstacle-free area detection results **a) (left)**: Ongoing traffic within the detected lanes **b) (right)**: Incoming traffic outside of detected lanes.

ratio exceeds threshold O_t , the lane is assumed to be blocked by an obstacle at that row and the distance of the obstacle is recorded for that lane. The formal description of this algorithm is given in 2.

Algorithm 2: Obstacle-Free Zone Detection Algorithm.

```

O initialized to 0
for all  $O(u, v)$  do
  Compute 3D coordinates of the point in the
  stereo reference frame using  $I_d$  and (15)
   $\text{Dist} \leftarrow aX + bY + (c + \delta\lambda)Z - d$ 
  if  $\text{Dist} > O_t$  then
     $Y_g \leftarrow (d - aX - (c + \delta\lambda)Z) / b$ 
    Compute  $(u', v')$  using (17)
     $O_{(u', v')} \leftarrow O_{(u', v')} + 1$ 
  end if
end if
end for
for all rows  $i$  in  $O$  do
  for all lanes do
    if lane ratio  $\gamma > O_t$  and lane still open then
      Output the lane as a blocked lane at corre-
      sponding distance
    end if
  end for
end for

```

5 EXPERIMENTAL RESULTS

We applied this approach to a set of sequences recorded from an instrumented experimental vehicle (Beauchemin et al., 2010). The implementation of the technique executes at 15Hz, including the stereo depth computation, ground plane detection, particle filtering for lane detection, and obstacle-free area estimation. Thirty initial particles for the particle swarm were used, and the stereo image size was 320 by 240 pixels.

The experiments subjectively demonstrate that the algorithm is robust to occlusion and partially worn-off or occluded lane markers and various urban artifacts. As observed, our technique remains stable, even for some frames without any evidence of lane markers,

which is very difficult for most of the existing lane detection approaches. Even in the presence of significant lane marker occlusions, our approach still properly detects lanes.

To our knowledge, vehicular imagery with annotated lanes and precise GPS data for the recording vehicle do not exist at this time, preventing an empirical evaluation of our algorithms. Among our short-term objectives is to produce such annotated sequences for comparative purposes. However, problems such as precisely determining the GPS position of the experimental vehicle for such sequences remain elusive and need to be surmounted.

One may argue that the requirement for GPS-addressable, lane-annotated maps limits the areas in which this approach may be used, which is correct. However, we believe this approach can be used in most driving situations, so long as lane-annotated maps are automatically generated and made available to instrumented vehicles. Additionally, the confidence measure obtained from thresholding the likelihood function may be used to assess the reliability of detected lanes.

6 CONCLUSIONS

We proposed a map-based lane detection and obstacle-free area detection using lane-annotated maps, particle filtering, and stereo depth maps. Our main contribution consists of our lane model obtained from lane-annotated maps, allowing us to represent irregular, opening, and closing lanes that are often ignored in the current literature. Ironically, these types of lanes are crucially important for iADAS as they occur in critical areas such as intersections and merging and turning areas which constitute perilous zones. Our approach uses a robust model that does not entirely depend on an on-board imaging system which may at times lead astray by the presence of occluding obstacles and worn-off lane markers.

REFERENCES

- Beauchemin, S., Bauer, M., Laurendeau, D., Kowsari, T., Cho, J., Hunter, M., and McCarthy, O. (2010). Roadlab: An in-vehicle laboratory for developing cognitive cars. In *23rd International Conference on Computer Applications in Industry and Engineering*, pages 7–12.
- Borkar, A., Hayes, M., Smith, M. T., and Pankanti, S. (2009). A layered approach to robust lane detection at night. In *IEEE Workshop on Computational Intel-*

- Intelligence in Vehicles and Vehicular Systems*, pages 51–57.
- Catmull, E. and Rom, R. (1974). A class of local interpolating splines. *Computer-Aided Geometric Design, Academic Press, New York*, pages 317–326.
- Cheng, H.-Y., Jeng, B.-S., Tseng, P.-T., and Fan, K.-C. (2006). Lane detection with moving vehicles in the traffic scenes. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):571–582.
- Hillel, A. B., Lerner, R., Levi, D., and Raz, G. (2012). Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, pages 1–19.
- Huang, A. S., Moore, D., Antone, M., Olson, E., and Teller, S. (2009). Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, 26(2):103–122.
- Jiang, Y., Gao, F., and Xu, G. (2010). Computer vision-based multiple-lane detection on straight road and in a curve. In *IEEE International Conference on Image Analysis and Signal Processing*, pages 114–117.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948.
- Kim, Z. (2008). Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):16–26.
- McCall, J. C. and Trivedi, M. M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37.
- Narita, Y., Katahara, S., and Aoki, M. (2003). Lateral position detection using side looking line sensor cameras. In *Intelligent Vehicles Symposium*, pages 271–275.
- Nieto, M., Salgado, L., Jaureguizar, F., and Arróspide, J. (2008). Robust multiple lane road modeling based on perspective analysis. In *15th IEEE International Conference on Image Processing*, pages 2396–2399.
- Pomerleau, D. (1995). Ralph: Rapidly adapting lateral position handler. In *Proc. Intelligent Vehicles Symposium*, pages 506–511.
- Rasmussen, C. and Korah, T. (2005). On-vehicle and aerial texture analysis for vision-based desert road following. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 66–66.
- Samadzadegan, F., Sarafraz, A., and Tabibi, M. (2006). Automatic lane detection in image sequences for vision-based navigation purposes. In *Proceedings of the ISPRS Commission, 5th Symposium on Image Engineering and Vision Metrology*.
- Sawano, H. and Okada, M. (2006). A road extraction method by an active contour model with inertia and differential features. *IEICE Transactions on Information and Systems*, 89(7):2257–2267.
- Talbi, E.-G. and Muntean, T. (1993). Hill-climbing, simulated annealing and genetic algorithms: a comparative study and application to the mapping problem. In *IEEE Proceedings of the 26th International Conference on System Sciences*, volume 2, pages 565–573.
- Watt, A. and Watt, M. (1991). Advanced rendering and animation techniques: Theory and practice. *Reading, MA*.
- Wu, S.-J., Chiang, H.-H., Perng, J.-W., Chen, C.-J., Wu, B.-F., and Lee, T.-T. (2008). The heterogeneous systems integration design and implementation for lane keeping on a vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):246–263.
- Zhou, Y., Hu, X., and Ye, Q. (2005). A robust lane detection approach based on map estimate and particle swarm optimization. In *Computational Intelligence and Security*, pages 804–811. Springer.