Tracking and Estimation of Ego-Vehicle's States for Lateral Localization

Young-Woo Seo

CMU-RI-TR-14-08

May 2014

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

For urban driving, knowledge of ego-vehicle's position is a critical piece of information that enables advanced driver-assistance systems or self-driving cars to execute safety-related, autonomous driving maneuvers. This is because, without knowing the current location, it is very hard to autonomously execute any driving maneuvers for the future. The existing solutions for localization rely on a combination of Global Navigation Satellite System (GNSS), an inertial measurement unit, and a digital map. However, on urban driving environments, due to poor satellite geometry and disruption of radio signal reception, their longitudinal and lateral errors are too significant to be used for an autonomous system. To enhance the existing system's localization capability, this work presents an effort of developing a vision-based lateral localization algorithm. The algorithm aims at reliably counting, with or without observations of lane-markings, the number road-lanes and identifying the index of the road-lane on the roadway that our vehicle happens to be driving on. Testings the proposed algorithms against inter-city and inter-state highway videos showed promising results in terms of counting the number of road-lanes and the indices of the current road-lanes.

Contents

1 Introduction

Knowledge of ego-vehicle's position is a critical piece of information for advanced driver-assistance systems (ADASs) or self-driving cars to execute safety-related, autonomous driving maneuvers. For example, to alert human drivers about lane departures, an ADAS needs to know a vehicle's position within the road-lane that the vehicle happens to be driving on [8, 9]. To avoid an impending collision, a vehicle may take over its control from a human driver to swerve to a neighboring lane if a braking distance is too short to achieve. Such an automatic control would be very hard to achieve, if the vehicle's location is unknown. The problem of identifying a vehicle's location has been tackled by using a combination of GNSS (e.g., GPS for U.S., GLONASS for Russian federation, Galileo for the European union), an inertial measurement unit, and a digital map (e.g., "NavTeq"). Most of the existing GNSS receivers used in consumer cars, however, using a single-channel antenna, provides only roughly accurate information about vehicles' longitudinal and lateral locations. Their location estimates are about 10 to 30 meters off from road's driving direction (i.e., longitudinal direction) and are not precise enough to tell which road-lane a vehicle is currently driving on (i.e., lateral direction) [7, 12, 16]. On urban driving environment, due to poor satellite geometry and disruption of radio signal reception by urban structures, their accuracies get worse, and result in unreliable and inconsistent provision of vehicle's location information. To overcome such challenges of localization problems at urban streets, many researchers, with an eye on cheaper costs and installation flexibility, have studied vision sensor usages. In this paper, we presents an effort of developing a computer vision algorithm that analyzes a stream of images acquired from a forward-looking, monocular camera to extract information for lateral localization. Particularly, this algorithm aims at producing information regarding 1) the number of road-lanes, 2) the index of a road-lane and 3) the lateral distances of our vehicle to the left and right boundaries of the road-lane that our vehicle happens to be driving on. To do so, our algorithm first detects longitudinal, lane-markings and represents them on road-plane coordinates. Our algorithm then uses the detected lane-markings as measurements for a Bayes filter to track the vehicle's lateral locations over frames. And finally our algorithm solves a homography between the image plane and the road plane to compute the actual metric distances within in a road-lane.

The pipeline of our algorithm begins with a detection of longitudinal lane-markings. Many excellent works have been done in the area of lane-marking detection. We refer to [8, 9] for a comprehensive literature survey and here we briefly discuss only the work relevant to ours. To detect longitudinal lane-markings, some investigated lane-markings' appearances (e.g., regularity in shapes [14] and homogeneity in color [4]). Others including ours have utilized the fact that there are intensity contrasts between lane-marking pixels and their neighboring pixels [1, 3, 10]. However, because other objects appearing on input images of urban street scenes (e.g., skid-marks, lane-marking patches, parts of vehicles) might exhibit such intensity contrasts, the performances of these approaches can be degraded due to many false positives. Other methods have used extra information, such as geometric structures of road lanes or road scenes, to improve lane-marking detection results [17, 19]. Similarly, we utilize the result of vanishing point detection to improve an initial result of our intensity-contrast based

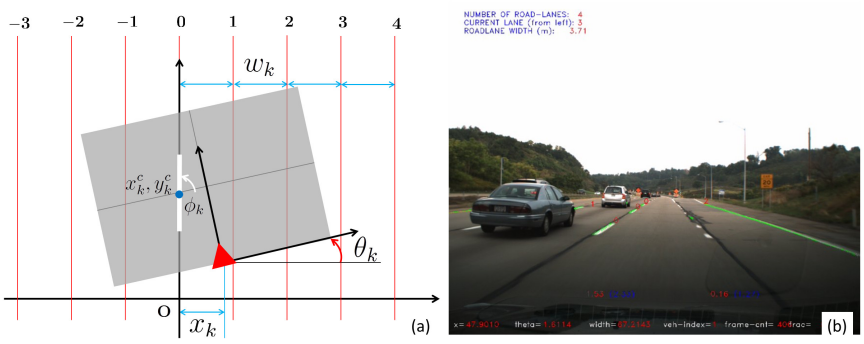

Figure 1: (a) A model of a road-plane. (b) An example output of lateral localization. At the top-left, the information about ego-vehicle's lateral location is displayed. The green pixel blobs are the detected lane-marings. The red numbers on the green pixel blobs represent lane-marking group identifiers.

lane-marking detection [18].

The task of vehicle localization is to estimate a vehicle's position with respect to landmarks on a map built a priori. Such a localization information is a prerequisite for an ADAS or a self-driving car to provide intelligent and autonomous driving maneuvers. Because of its importance, the vehicle localization at urban driving environments has been extensively studied for the past decades [12, 5, 11, 13, 6]. These localization methods are different from one another in terms of the characteristics of maps, sensor measurements, methods of estimation and tracking vehicles' states. Most work estimated vehicles' locations by fusing sensory measurements from multiple sensors in different modalities and tracked the estimated locations over time [5, 11, 12, 13]. For example, Tao et al. used a commercially available lane-departure warning system for correcting the location estimation based on GPS and inertial measurements [13]. Similarly, Rose [12] employed an extended Kalman filter to fuse the detected lane-markings with inertial measurements for lane-level vehicle localization. Our work is closely related to these two works [12, 13], but is different in that our work focuses on solving the problem of lateral localization without using any maps.

Our contributions include 1) the development of a novel representation of road geometry, 2) the enhancement of longitudinal lane-marking detection using the result of a vanishing point detection, 3) the development of a Bayes filter for tracking ego-vehicle's state, and 4) the empirical validation of the proposed method's usefulness.

In what follows, we detail a model of a road-plane, a method for detecting longitudinal lane-markings in Section 2.1, a Bayes filter for tracking our vehicle's lateral location in Section 2.2, and, in Section 2.3, a solution for the homography between image plane and road plane to compute the world coordinates of road-plane's location. We then discuss experiments and their findings in Section 3.

2 A Computer Vision System for Lateral Localization

To accomplish the goal of vision-based lateral localization, we first detect longitudinal lane-markings and use them as measurements to estimate ego-vehicle's lateral location. We assume that road is flat and develop a road plane model to represent multi-lane roads. Figure 1 (a) illustrates our road-plane model. There are nine red, vertical lines numbered from -3 to 4 that depict boundaries of road-lanes. Each red line corresponds to a longitudinal, lane-marking and its painting is determined by its traffic-control function, either dashed or solid in different colors (e.g., yellow, white or blue) [15]. We assign the index "0" to the lane-markings detected from the very left to our vehicle and increase (or decrease) the number based on the relative distance from our vehicle image coordinates. The range of these indices is determined by the setup (e.g., CCD size and lens) of our vision sensor – our lane-marking detector is capable of detecting up to four lanes in the same driving direction. For example, suppose our vehicle is driving on the rightmost lane of a four road-lanes highway, then the vertical lines between "-3" and "1" will be the leftmost and rightmost boundaries of four road-lanes. A gray rectangle represents an image plane where a red triangle represents the camera location in the image plane (i.e., the image coordinates of vehicle center) and a white thick line represents a detected longitudinal lane-marking. A detected lane-marking is represented by a triplet of the image coordinates of its centroid and orientation, $lb_l = [x_l, y_l, \phi_l]^T$. In our road-plane model, the vehicle's lateral location is represented by 1) the index of the road-lane that the vehicle happens to be driving on and 2) the lateral offsets from the left and the right boundaries of that road-lane. Figure 1 (b) shows an example of the output image where the detected lane-markings are depicted in green. At the top left, our algorithm outputs the information about our vehicle's lateral location: the number of road-lanes (i.e., 4) and the index of the current road-lane (i.e., 3) where the index of a road-lane begins to count from the leftmost. At the bottom of the same figure, the lateral distances to the left and right boundary of the current road-lane are computed.

2.1 Longitudinal Lane-marking Detection

We detect lane-markings by applying a simple filter to an input image. The filter is designed to identify the intensity contrast between lane-marking pixels and their neighboring pixels [10]. We define a region of interest (ROI) for lane-marking detection and apply this spatial filter to the ROI, to obtain a new intensity image about lane-markings. An example of such an intensity image is shown at Figure 2 (a). The rectangle in a yellow-dashed line represents the boundary of the ROI. To identify a set of lane-marking blobs, we first do an intensity thresholding to this new intensity image, to produce a binary image of lane-markings. We then apply a connected-component grouping, resulting in a number of lane-marking, pixel blobs. To represent each pixel blob as a line segment, we compute the eigenvalues and eigenvectors of the pixel coordinates' dispersion matrix. The eigenvector, $\mathbf{e}_1$, associated with the largest eigenvalue is used to represent the orientation of a line segment and its length, $\mathbf{l} = (\phi, \rho) = (atan2(\mathbf{e}_{1,2}, \mathbf{e}_{1,1}), \bar{x}\cos\phi + \bar{y}\sin\phi)$, where $\bar{x} = \frac{1}{n}\Sigma_i x_i, \bar{y} = \frac{1}{n}\Sigma_i y_i$. A pixel blob, lb_l, is then represented as a triplet of the coordinates of its centroid, x_l

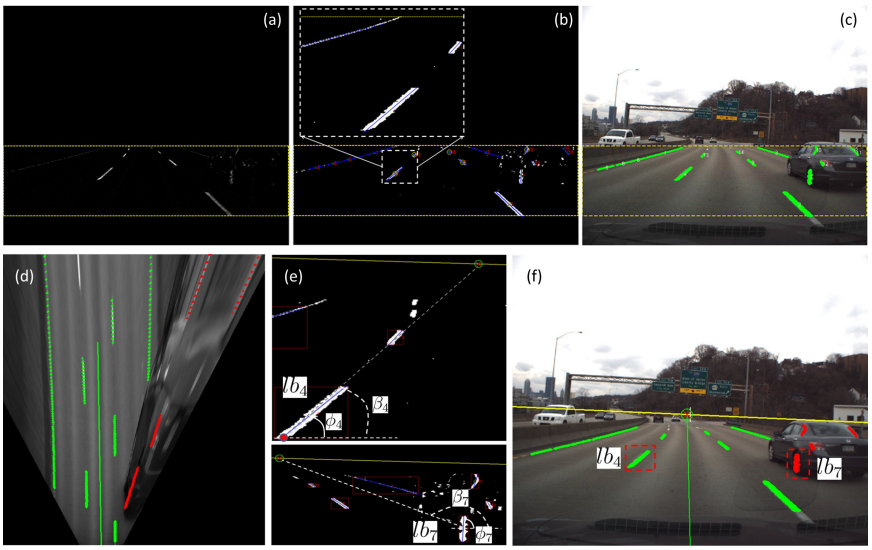

Figure 2: Examples of longitudinal, lane-marking detection results. (a) A spatial filtering results in a new intensity image about lane-markings. (b) An intensity thresholding produces a binary image about lane-markings. (c). A example of the lane-marking detection before removing false-positives. (d) The green line depicted at the center of image illustrates the approximated, using the result of a vanishing point detection, instantaneous driving direction. (e) Removal of the erroneously detected lane-markings. (f) An improved lane-marking detection results.

and y_l, and its orientation, ϕ_l. To fit a line to a pixel blob, we tried three methods: the line-fitting based on eigen-analysis, the probabilistic, and the standard Hough transform. We found the eigen-analysis method to work best in terms of the number of resulting lines and representation fidelity to the patterns of low-level features. Figure 2 (b) shows examples of the identified blobs in a binary image. We remove some lane-marking blobs if their lengths are too short or long. Figure 2 (c) shows an example of the lane-marking detection results. The outputs of our lane-marking detector are useful because their false negatives are, depending on the intensity thresholding, very small. At the same time, however, the number of false positives increase because there are other objects on the input images that have the same intensity contrast. For example, in Figure 2 (c), four parts of a car have such intensity contrasts, that satisfy the intensity contrast, resulting in being incorrectly detected lane-markings.

To remove such false positive detections in a principled way, we use the result of vanishing point detection. We detect, using the extracted line segments, the vanishing point on a horizon. On Figure 2 (f), the green circle with red "X" mark shows the detected vanishing point and the yellow, slanted line represents the estimated horizon line [18]. The line between the image coordinates of the image bottom mid-point (i.e., the image coordinates our camera is projected on) and the vanishing point on a horizon line is used to approximate an instantaneous driving direction of the road that our

vehicle happens to be driving on. This is an instantaneous approximation because it linearly approximates a polynomial driving direction at the location where the input image is acquired. The green vertical line in Figure 2 (f) shows an approximated road driving direction in a perspective image. Figure 2 (d) in fact clarifies the idea of linearly approximating the driving direction where the detected lane-markings, the results of driving direction approximation and the image subregions defined by the ROI are projected on an inverse perspective image. Our idea is then to filter out some of the lane-marking blobs if their orientations are not aligned with this instantaneous driving direction. We do this because the orientations of any true, longitudinal lane-markings should be aligned with this instantaneous driving direction. Two figures at Figure 2 (e) shows an example of applying this idea to the initial result of the lane-marking detection. We first compute the difference between a blob initial orientation (e.g., β_7) and a new orientation linking the blob's mid-point to the detected vanishing point. And then we remove the blob under investigation (e.g., lb_7) if the computed orientation difference is greater than a predefined threshold (e.g., 20 degrees). By executing these two procedures, our method is able to remove those four false positive detections on the car. Figure 2 (f) shows the refined output of the longitudinal, lane-marking detection, where the red-blobs are the false-positive lane-marking detection that are removed from the final results.

The last step of the lane-marking detection is to determine the class of a detected lane-marking based on its color and shape. We trained a random-tree classifier [2] to assign a detected lane-marking with one of the predefined color classes, $\{$White, Yellow, Other$\}$. For the lane-marking's style classification, we consider the length of a detected lane-marking and assign one of the predefined style classes, $\{$Solid, Dashed, Other$\}$. We have "Other" class for both classification to handle a class other than the main target classes (e.g., "White" and "Yellow").

2.2 Tracking and Estimating of Ego-Vehicle's Lateral Location

The previous section describes how our method detects longitudinal, lane-markings. In this section, we detail how our method tracks, using the detected lane-markings, ego-vehicle's state and compute, using the tracked state, lateral location.

As illustrated in Figure 1 (a), we define the state of ego-vehicle at time step k as $\mathbf{x}_k = [x_k, \theta_k, w_k]^T$, where x_k is the distance from the ground coordinates' origin, θ_k is the orientation difference between the vehicle's driving direction and that of the ground, w_k is the width of road-lane in pixels. We develop an unscented Kalman filter (UKF) for modeling a non-linear state transition of our vehicle's lateral location. Our filter begins with predicting what the next state will be, based on the following process model.

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} x_{k+1} \\ \theta_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + v \sin\theta_k dt \\ \theta_k + \omega dt \\ w_k dt \end{bmatrix} \tag{1}$$

where v is a speed, and ω is a yaw rate, and dt is a discretization unit for a numerical integration.

At a predicted state $\hat{\mathbf{x}}_k^-$, a measurement model is used to predict an expected measurement. For our case, a measurement is a lane-marking, $lb_l = \mathbf{z}_l$. We define the measurement model as:

$$\hat{\mathbf{z}}_k = h(\mathbf{x}_k) = \begin{bmatrix} \hat{\phi} \\ \hat{x} \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} \pi/2 - \phi_l \\ \frac{1}{\cos\phi_l}(lb_{index} - x_l + y_l \sin\phi_l) \end{bmatrix} \tag{3}$$

where $[x_l, y_l, \phi_l]^T$ is the coordinates of the lth detected lane-marking's centroid and orientation. The measurement model aims to transform these values of a detected lane-marking in the image coordinates into ones ($\hat{\phi}$,$\hat{x}$) in the road-plane so as to use them for ego-vehicle's state update. $lb_{index} \in \{-3, -2, -1, 0, 1, 2, 3, 4\}$ is the lane-marking group index of the detected lane-marking shown at Figure 1 (a). To compute a lane-marking's group index, we first, to determine which side of our vehicle a lane-marking is detected, compute a cross-product between a detected lane-marking's centroid and two end-points of the approximated driving direction line. The lane-marking is located at left (right) to our vehicle if the determinant of the cross-product is greater (less) than zero. With this relative location and two variables from the current state, x_k and w_k, the lane-marking's index is computed by

$$lb_{index} = \begin{cases} round(idx + 0.5) & \text{if } idx \geq 0 \\ round(idx - 0.5) & \text{Otherwise} \end{cases} \tag{4}$$

where $idx = \frac{\pm d_k + x_k}{w_k}$, d_k is a lateral distance between a detected lane-marking lb_l and the vehicle image coordinate, and $-d_k$ (d_k) if lb_l is found from left (right). In a state update step, the state, $\hat{\mathbf{x}}_k$ (and its covariance matrix $\mathbf{P}_k$) is estimated by investigating the difference between the expected measurement, $h(\hat{\mathbf{x}}_k^-)$ and the actual measurement, $\mathbf{z}_l$ (i.e., a detected lane-marking), $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_l - \hat{\mathbf{z}}_k)$, ($\mathbf{P}_k = \mathbf{P}_k^- - K_k \mathbf{P}_z K_k^T$). Where K is the Kalman gain that determines how much the innovation $(\mathbf{z}_l - \hat{\mathbf{z}}_k)$ is used to compute the estimate, $\hat{\mathbf{x}}_k$.

Given an estimated ego-vehicle's state $\hat{\mathbf{x}}_k$, for the lateral localization, we 1) determine, based on their color and style classification, which of the detected lane-marking groups should be the left and right boundaries of the road, 2) count the number of road-lanes, and 3) compute, using the tracked state the index of the current road-lane. For example, in Figure 1 (b), the lane-marking groups labeled "-2" and "2" are determined as the left and the right boundary of the road based on the color and style classification, i.e.x) "Yellow" and "Solid" for the lane-marking group "-2" and "White" and "Solid" for the lane-marking group "1." Given the result of this computation that there are five longitudinal lane-markings and thus four road-lanes, our algorithm analyzes the value of x_k (i.e., 47.9010) to estimate the index of our vehicle, i.e.) the value of x_k indicates that our vehicle drives between the lane-marking group "0" and "1." However, such a logic may fail to work when lane-markings are not clearly visible at a given time. At urban driving environment, lane-markings may not be visible due to neighboring cars, poor quality of lane-marking paintings, bad weather, etc. When this happens, the algorithm may fail to correctly count the number of road-lanes, resulting in incorrect computation of the index of the current road-lane. To handle with such cases, we

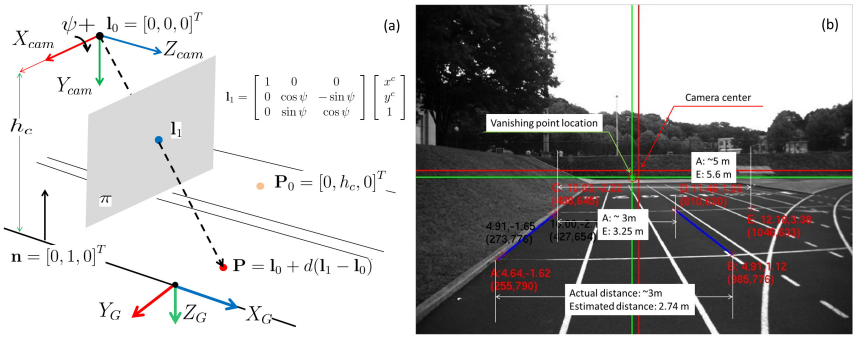

Figure 3: (a) A model of the homography between the image and ground planes. (b) An experimental setup to verify the accuracy of our homography model.

create, to record a history of lane-marking detections, a matrix, $H = T \times 9$ matrix, where $T = 1, ..., t, ..., |T|$ and initializes its value with -1. A column corresponds to a longitudinal lane-marking (or a lane-marking group) and a row corresponds to a time step. Specifically, the algorithm records what type of lane-markings are observed at which lane-marking groups in the time step t, at the tth row with the values 0 for "Yellow-Dashed," 1 for "Yellow-Solid," 2 for "Yellow-Other," 3 for "White-Solid," 4 for "White-Dashed," and 5 for "White-Other." The row of the oldest history is replaced with the newest history about lane-marking detection and its content is exponentially decayed.

2.3 Metric Measurement

To know where our vehicle is in a road-lane, we need to compute the lateral distances (in meter) of our vehicle to the left and the right boundaries of a road-lane. To do so, it is necessary to compute the world coordinates of points on the road plane. To this end, we solve a homography between an image plane and a road plane. Figure 3 (a) illustrates the homography model between the image, π, and the road plane. We assume that there are no roll and yaw, but pitch, ψ between the road-plane and the image-plane. We represent, using a parametric line equation, the geometric relation between a point, $\mathbf{l}_1$ on an image plane and its projection, $\mathbf{P}$ and solve this equation to compute the coordinates of the point on the ground.

$$\mathbf{P} = \mathbf{l}_0 + d(\mathbf{l}_1 - \mathbf{l}_0) = d\mathbf{1}_1 + \mathbf{l}_0 = d\mathbf{1}_1, \tag{5}$$

$$= \left[h_c \frac{x_{cam}}{y_{cam}\cos\psi + \sin\psi}, h_c, h_c \frac{-y_{cam}\sin\psi + \cos\psi}{y_{cam}\cos\psi + \sin\psi} \right]^T \tag{6}$$

where $d = \frac{(\mathbf{P}_0 - \mathbf{l}_0)\cdot\mathbf{n}}{\mathbf{l}_1\cdot\mathbf{n}} = \frac{\mathbf{P}_0\cdot\mathbf{n}}{\mathbf{l}_1\cdot\mathbf{n}} = \frac{h_c}{y_{cam}\cos\psi - \sin\psi}$ and $\mathbf{l}_1 = \begin{bmatrix} x_{cam} \\ y_{cam}\cos\psi + \sin\psi \\ -y_{cam}\sin\psi + \cos\psi \end{bmatrix}$.

Figure 3 (b) shows a setup for verifying the accuracy of our homography between the image and the ground plane. We manually measure the distances between the camera

and markers on the ground to evaluate the accuracy of the pitch angle computation. We found that the distance measurements have, on average, a sub-meter accuracy (i.e., less than 30 centimeter).

3 Experiments

This section details the settings and the results of the experiments that we carried out to evaluate the performance of the proposed algorithm. The goal of this work is to develop a computer vision algorithms that produces information for lateral localization, such as 1) the index of the current road-lane, and 2) the lateral distances of our vehicle to the left and right boundaries of the current road-lane.

To collect data including videos and information about vehicle's motion, we drove our robotic vehicle on a route about urban streets, inter-city, and inter-state highways. Our vehicle is equipped with a high-end navigation system (i.e., Applanix LV) that, in root-mean-square sense, the error of pitch angle estimation is 0.02 with GPS signals (with RTK corrections) or 0.06 degree with GPS outage, when driving more than one kilometer or for longer than one minute. The vision sensor installed on our vehicle is a PointGrey's Flea3 Gigabit camera, which can acquire an image frame of 2448×2048, maximum resolution at 8Hz. For a faster, real-time processing, we rescaled the original resolution into half, detected line segments and longitudinal, lane-markings from a predefined ROI, $x_1 = 0$, $x_2 = \mathbf{I}_{width-1}$, $y_1 = 1300$ and $y_2 = 1800$.[1] We used such a high-resolution camera not just for lateral localization, but also for moving object detection and tracking and for traffic light detection. Thus each vision algorithm needs to rescale the original resolution of input images to guarantee of a real-time response. For producing a binary image of lane-markings, we set the intensity threshold 10. We implemented the proposed algorithm in C++ with OpenCV libraries. Our implementation of longitudinal, lane-marking detection ran about 7Hz and the vanishing point detection 8Hz on a 2.7 GHz quad core. While driving the route, we ran a data logger for collecting images and information about vehicle's motion. When we replay the log or evaluate the proposed algorithms with the log, the data logger automatically synced the high-rate, motion data (i.e., 100Hz) with the low-rate, image data (i.e., 8Hz). For the UKF, we used the Bayes++ package[2] to implement our filter and initialized the state and its error (or covariance) matrix, $\mathbf{x}_0 = [x_0, \theta_0, w_0]^T = [50, 0, 100]^T$ and $\mathbf{P}_0 = diag\left([10^2, 0.1^2, 10^2]\right)$, where the values of x and w are in pixels and that of θ is in radian. In addition, the noises of the process model, $\mathbf{Q} = diag\left([5^2, 0.1^2, 5^2]\right)$ and the measurement model, $\mathbf{R} = diag\left([0.1^2, 5^2]\right)$.

We tested the proposed algorithm with seven video data about urban streets, inter-city, and inter-state driving. All of the testing results are available from the web: the

[1] These y-values are in the original resolution. If the image is scaled to a half of the original, these y-values are scaled as well.

[2] `http://bayesclasses.sourceforge.net/Bayes++.html`

	1	2	3	4	5	6	7	
Images	343	302	163	700	500	1,000	368	3,376
Correct	333	284	80	539	332	650	305	2,523
MSE	0.05	0.30	0.37	0.09	0.07	0.15	0.11	0.16

Table 1: Summary of the experimental results.

video 1[3], 2[4], 3[5], 4[6], 5[7], 6[8], and 7[9]. Table 1 summarizes the experimental results. The first row represents the number of images in each video data; the second row is about the number of images that our algorithm correctly counted the number of road-lanes; the last row is about the mean-squared error of the road-lane width estimation. To the best of our knowledge, no prior work about vision-based lateral localization is publicly available and thus no comparison with existing work. While recording each video, our algorithm outputs the number of road-lanes, the index of the current road-lane, the estimation of the current road-lane's width, and the lateral distances of our vehicle to the left and right boundary. Later, human annotators manually assigned individual videos with the number of road-lanes and the index of the road-lane that our vehicle happened to be driving on. For example, in the first video, there were 333 out of 343 image frames that our algorithm correctly counted the number of road-lanes. On the Table 1, we reported only the outputs of counting the number of road-lanes. This is because we found that the computations of the current road-lane's index were always correct when the number of road-lanes was correctly computed. To evaluate the road-lane width estimation and the lateral distance computation, we did our best to drive our vehicle at the center of the road-lane to utilize a nominal road-width in the U.S. This is because no ground truths were available for the routes – our vehicle does not have any sensors looking down to detect lane-markings for lateral distance computation. Thus, to evaluate the accuracy of our algorithm's road-lane with estimation, we used the nominal road-width recommended by the U.S. traffic authority [15] (i.e., 12 feet=3.65 meters), even though the widths of some testing routes (e.g., a ramp) are wider than that of the nominal lane. The mean-squared error of the road-lane width estimation was 0.05 meter.

For most of the video data, the states of the lane-marking paintings were somewhat obsolete, but generally in fair conditions for a detection. On about 30% of the images, lane-markings were partially and completely occluded by vehicles passed by our vehicle. As we recorded the videos without color calibration, the images acquired while passing overpasses were washed out. On overall, for the (inter-city and inter-state) highway driving data (i.e., the video 1, 2, 4, and 7), our algorithm showed a promising performance in that it correctly computed the number of road-lanes over 85% and com-

[3] https://www.youtube.com/watch?v=WmCCKCUX070
[4] https://www.youtube.com/watch?v=ommbRxYlmS8
[5] https://www.youtube.com/watch?v=2wj5aHOOlLA
[6] https://www.youtube.com/watch?v=0VjuhsrZNKs
[7] https://www.youtube.com/watch?v=Q7_C2cJRXco
[8] https://www.youtube.com/watch?v=LJ5va0ALEJg
[9] https://www.youtube.com/watch?v=f1i-dCZXRAw

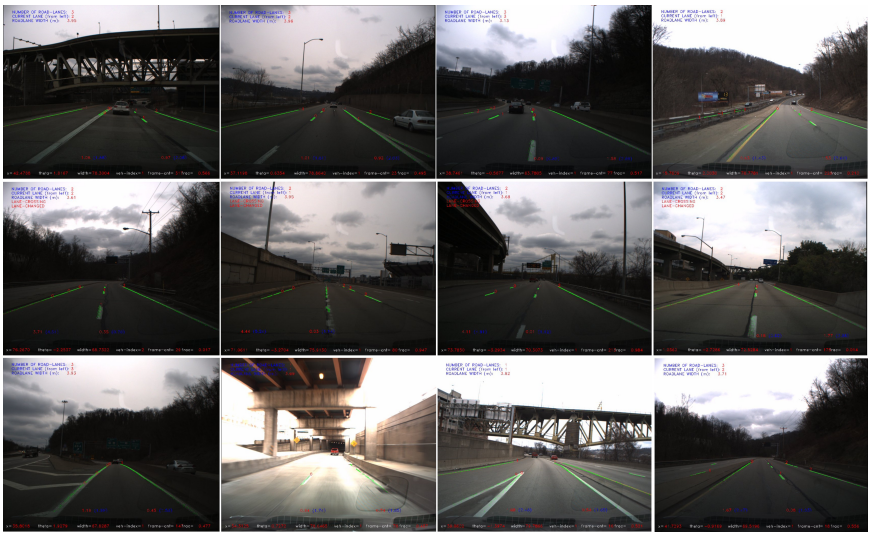

Figure 4: Experimental results of a route of inter-city and inter-state highway driving are shown.

puted the width of a road-lane with smaller error (i.e., less than 0.1 meter). Figure 4 shows some example outputs. The first and second rows show examples that our algorithm worked well. In particular, the images at the second row show examples that our algorithm was able to detect lane-changing maneuvers. Such a capability of recognizing lane-changing maneuvers would be very useful to correct incorrect localization if a map is available. The performance of our algorithms degraded when the frequency of the occluded lane-markings was high and the qualities of lane-marking paintings were low. This is because there is no history of road geometry to remember and re-use if 1) the road-geometry is changed frequently and 2) the traffic is high. For example, for the video data 3, the vehicle drove on a ramp to merge with a road with multiple-lanes. At the end of the ramp, there were four lanes, immediately after this, four lanes became 3 lanes, and then ended up two road-lanes. In addition to such a complex road geometry, the vehicle drove diagonally, over a short period of time, to change its lane from the leftmost to the rightmost. This challenging driving scenario and maneuver caused our algorithm fail to correctly identify the index of the current road-lane and estimate the widths of the road-lanes.

4 Conclusions and Future Work

This paper presents an effort of developing a computer vision algorithm that identifies our vehicle's lateral location. To do so, our algorithm first detects, using a spatial filter, longitudinal, lane-markings and uses the results of a vanishing point detection to enhance the lane-marking detection results. To reliably and consistently estimate our vehicle's lateral location, we developed a model of road-plane and in this model, our

algorithm, using an unscented Kalman filter, tracks and estimates our vehicle's lateral location. To handle with the case where no lane-markings are observed, we maintained a matrix of lane-marking detection history. Through testings with a collection of images acquired from driving on streets, inter-city, and inter-state routes, our algorithm demonstrated promising results in terms of counting the number of road-lanes and of identifying the index of the current road-lane.

As future work, we would like to determine the limits of our algorithm and so continue testing it against various driving environments. Particularly, we would like to test our algorithm against 1) urban and rural streets, 2) dusk and night driving scenarios, and 3) under raining and snowing.

References

[1] M. Aly. Real time detection of lane markers in urban streets. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 7–12, 2008.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[3] A. Broggi, A. Cappalunga, C. Caraffi, S. Cattani, S. Ghidoni, P. Grisleri, P. P. Porta, M. Posterli, and P. Zani. Terramax vision at the urban challenge 2007. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):194–205, 2010.

[4] K.-Y. Chiu and S.-F. Lint. Lane detection using color-based segmentation. In *IEEE Intelligent Vehicles Symposium*, pages 706–711, 2005.

[5] J. Du and M. J. Barth. Next-generation automated vehicle location systems: positioning at the lane level. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):48–57, 2008.

[6] C. U. et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics: Special Issues on the 2007 DARPA Urban Challenge*, 25(8):425–466, 2008.

[7] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Interscience, 2007.

[8] A. B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, 2012.

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

[10] M. Nieto, J. A. Laborda, and L. Salgado. Road environment modeling using robust perspective analysis and recursive bayesian segmentation. *Machine Vision and Applications*, 22:927–945, 2011.

[11] D. Obradovic, H. Lenz, and M. Schupfner. Fusion of sensor data in siemens car navigation system. *IEEE Transactions on Vehicular Technology*, 56(1), 2007.

[12] C. Rose. Lane level localization with camera and inertial measurement unit using an extended kalman filter. Master's thesis, Auburn University, 2010.

[13] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman. Mapping and localization using gps, lane markings and proprioceptive sensors. In *Proceedings of IEEE Conference on Intelligent Robots and Systems*, pages 406–412, 2013.

[14] L.-W. Tsai, J.-W. Hsieh, C.-H. Chuang, and K.-C. Fan. Lane detection using directional random walks. In *IEEE Intelligent Vehicles Symposium*, pages 303–306, 2008.

[15] F. H. A. U.S. Department of Transportation. *Manual on Uniform Traffic Control Devices for Streets and Highways*, 2009. http://mutcd.fhwa.dot.gov/.

[16] J. Wang, S. Schroedl, K. Mezger, R. Ortloff, A. Joos, and T. Passegger. Lane keeping based on location technology. *IEEE Transactions on Intelligent Transportation Systems*, 6(3):351–356, 2005.

[17] Y. Wang, E. K. Teoh, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision Computing*, 22:269–280, 2004.

[18] Young-Woo and R. R. Rajkumar. Utilizing instantaneous driving direction for enhancing lane-marking detection. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 170–175, 2014.

[19] S. Zhou, Y. Jiang, J. Xi, J. Gong, G. Xiong, and H. Chen. A novel lane detection based on geometrical model and gabor filter. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 59–64, 2010.