Use of a Monocular Camera to Analyze a Ground Vehicle's Lateral Movements for Reliable Autonomous City Driving

Young-Woo Seo and Ragunathan (Raj) Rajkumar

Abstract— **For safe urban driving, one prerequisite is to keep a car within a road-lane boundary. This requires human and robotic drivers to recognize the boundary of a road-lane and the location of the vehicle with respect to the boundary of a road-lane that the vehicle happens to be driving in. We present a new computer vision system that analyzes a stream of perspective images to produce information about a vehicle's lateral movements, such as distances from a vehicle to a road-lane's boundary and detection of lane-changing maneuvers. We improve existing work in this field and develop new algorithms to tackle more challenging cases, such as driving on inter-city highways. Tests on real inter-city highways showed that our system provides stable and reliable performance in terms of computing lateral distances, while yielding reasonable performance in detecting lane-changing maneuvers.**

I. Introduction

In city-driving scenarios, an essential component of safe driving is keeping the vehicle in a road-lane boundary. In fact, such a capability is a prerequisite for various advanced driver assistance systems (ADAS) [3], [5], [12] as well as for executing reliable autonomous driving [15], [20]. One way to achieve this capability, for human drivers, is to design lane-departure warning systems. By analyzing steering commands from in-vehicle data and lane-markings through a forward-looking camera, such a warning system can alert drivers when they unintentionally deviate from their paths. A self-driving car, to be deployed on urban streets, should be capable of keeping itself in a road lane before executing any other urban autonomous driving maneuvers, such as changing lanes and circumventing stalled or slow-moving vehicles.

The task of staying within a road-lane begins with perceiving longitudinal lane-markings. A successful detection of such lane-markings leads to the extraction of other important information – the vehicle's location with respect to the boundary of the road-lane. Such information about lateral distances of the vehicle to the left and right boundaries of a road-lane help a human driver and a robot driver keep the vehicle in the road-lane boundary. The capability of driving within designated lanes is critical for autonomous driving on urban streets, where GPS signals are either degraded or can be readily disrupted.

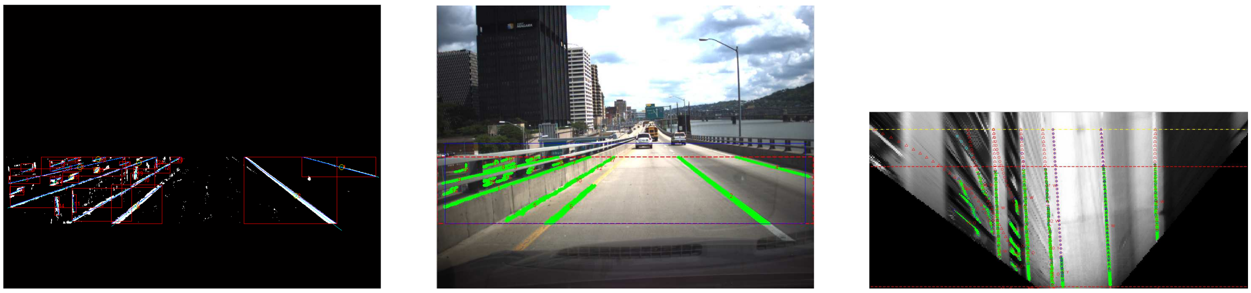
Some earlier work, using 3D LIDARs, demonstrated impressive results in understanding road geometry. In particular, four of the autonomous driving applications installed multiple off-the-shelf laser range finders toward the ground and measured the reflectivity values of road surfaces. In such manner they analyzed the geometry of the current roadway [6], [11], [15], [20]. Two of ADAS applications proposed lane-departure warning systems using automotive-grade laser range scanners. Instead of multiple LIDARs, they used a single LIDAR with multiple horizontal planes: six for Ogawa and Takagi [17] and four for Kibbel et al. [10]. Both methods recognized lane-markings in a similar way: 1) handpicking some of the scan points, 2) finding a list of parameters, (e.g., curvature, lane-width, lateral offset, and yaw-angle), and 3) representing the lane with a polynomial (e.g., quadratic or cubic).

However, such a high-end, expensive LIDAR may not always be available. Instead of relying on such active range sensors, many researchers as an alternative, with an eye on lower costs and installation flexibility, have studied the use of vision sensors. Researchers have actively studied road geometry understanding through lane-marking detection; some research results have been successfully commercialized as well [3]. Some utilize inverse perspective mapping to remove perspective distortions [1], [16], others use in-vehicle data, such as steering angle, velocity, whether a wiper is turned on [3], [12]. Some have implemented Bayes filters, to make their lane-detection methods robust [9], [10], [16], [17]. However, most of this research using a vision sensor focuses on developing driver-assistance systems for manual driving, where the outputs are not always expected to be produced and human drivers can, if necessary, override the incorrect outputs [1], [3], [5], [12], [16]. For a self-driving car, in contrast, the information about a vehicle's location with respect to a road-lane boundary should be available throughout navigation and in a bounded performance. Otherwise, when driving on regions with unreliable GPS signal reception (e.g., urban canyons), an autonomous vehicle might easily veer from the centerline of a road-lane, resulting in unacceptable consequences.

To produce a vehicle's relative motions within a road-lane, we develop a vision algorithm that analyzes perspective images acquired from a monocular camera to extract information about a vehicle's lateral movements, metric offset measurements from road-lane boundaries, and detection of lane-changing maneuvers. To this end, our algorithm first extracts longitudinal lane-markings from input perspective images and, on inverse perspective images, analyzes their geometric relation. This step yields the local geometry of a current roadway. The algorithms then solve a homography between a camera plane and a roadway plane to assign the identified geometry with metric information.

The contributions of this paper include 1) a method of analyzing the geometry of a current roadway, 2) a method of computing metric information of points on the ground plane, and 3) a new vision system for computing a vehicle's lateral movements.

Young-Woo Seo is with the Robotics Institute and Ragunathan Rajkumar is with Dept of Electrical Computer Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, young-woo.seo@ri.cmu.edu, raj@ece.cmu.edu



(a) An example of lane-marking detection results. Our lane-marking detector produces a binary image about lane-markings and the detected lane-markings are represented as a list of pixel groups (or blobs). Each of the red boxes shows a bounding-box of a detected lane-marking blob.

(b) The initial lane-marking detection results are overlayed onto, after false positive removal, the input image. The blue rectangle defines the image region that is transformed into an inverse perspective image.

(c) This subfigure shows only a part of an inverse perspective image to enlarge the image sub-region where lane-marking blobs appear.

Fig. 1: Results of a lane-marking detection.

II. Understanding Lateral Motions of A Ground Vehicle from A Single Image

Our goal in this work is to provide a ground vehicle with information about its lateral movements. We call the road-lane, which our vehicle happens to be driving on, the host road-lane. The information provided includes the vehicle's lateral location in meters relative to the host road-lane's boundary and occurrences of lane-changing maneuvers. To acquire such information, our vision algorithms first detect longitudinal lane-markings on the images acquired from a forward-looking camera, and classify their colors (e.g., yellow or white); then transform a perspective image into an inverse perspective image to obtain the information about the geometric structure of the host roadway, such as the number of road-lanes in the current roadway and the index of the host road-lane from the leftmost road-lane; and, finally, we compute metric measurements of the identified regions to obtain information about the vehicle's lateral motion.

In what follows, we detail how we recognize lane-markings from perspective images and compute the geometry of a local roadway from inverse perspective images. We then explain how we compute 3-dimensional world coordinates of 2-dimensional image coordinates of the identified roadway geometry so as to produce the information about the vehicle's lateral motion in meters.

A. Recognizing Lane-Markings for Understanding Local Geometry of Roadway

Road-markings define how drivable regions are used to guide vehicles' navigation. They are obviously important cues to understanding the geometric structure of a roadway. Among these, the ones we want to detect are those that longitudinally depict boundaries of individual road-lanes. In a forward-looking image of urban-streets, we can readily, with the naked eye, distinguish lane-markings. They have distinguishing colors (white and yellow), relatively higher intensity than their neighboring pixels, and occupy approximately known locations. However, these salient features are not always available for image processing; after all the actual values of lane-marking pixels vary based on image acquisition conditions.

Instead of dealing directly with these challenging variations in lane-marking pixels' appearances, we identify lane-marking image regions by implementing a simple filter, which emphasizes the intensity contrast between lane-marking pixels and their neighboring pixels. Our lane-marking detection algorithm was inspired by the one developed by Nieto and his colleagues [16].

Normal longitudinal pavement lane markings on highways (i.e., inter-city and inter-state highways in the U.S.) are 4∼12 inches wide (10∼30.48 centimeters) [13]. Given this fact, we can readily compute the number of pixels used to depict lane-markings on each row of the input image. For example, for a given pre-computed lane-marking pixel width, w_i, our filter transforms the original image intensity value, $I(u,v)$, into $I(u,v)'$ by

$$\begin{aligned} I(u,v)' &= 2 \times I(u,v) - \{I(u-w_i,v) + I(u+w_i,v)\} \\ &- |I(u-w_i,v) - I(u+w_i,v)| \end{aligned}$$

If $I(u,v)'$ is greater than a predefined maximum value, we set it to that maximum value (e.g, 255). If $I(u,v)'$ is lesser than zero, we set it to 0. To produce a binary image of lane-markings from this filter response, we do a thresholding that keeps only pixels of which values are greater than a given threshold. Figure 1a shows an example of lane-marking detection results. Even with many (readily discernible) false positive outputs, our lane-marking detection outputs are sufficient because their false negatives are quite small, meaning that our detector picked up almost all true longitudinal lane-markings appearing in the image. We then represent the lane-marking detection result as a list of pixel groups (or blobs) and analyze their geometric properties, such as heading and length, to filter out some non-lane-marking blobs. To further filter out false positives, we also compute the ratio of the sum of a blob's width to that of a true lane-marking to estimate the likelihood that a lane-marking blob is a true lane-marking.

$$\gamma(b_i) = \frac{\sum_{v_j} |u_{j,1} - u_{j,|u_j|}|}{\sum_{v^*_j} |u^*_{j,1} - u^*_{j,|u_j|}|}$$

where b_i is the ith lane-marking blob, $u_{j,1}$ ($u_{j,|u_j|}$) is the jth row's first (last) column of the $i$the blob, and v^*_j is the corresponding information of the true lane-marking blob.

The color of a lane-marking plays an important role of determining its semantics. For example, in the U.S., a yellow (or white) longitudinal lane-marking separates traffic flows in the opposite (same) direction [13]. To obtain such semantic information about a lane-marking, we classify, using a Gaussian mixture color model, the color of a detected lane-marking blob into one of three categories: yellow, white, and other. In particular, the color class of a detected lane-marking blob is determined by computing, $\arg\min_{c \in C}(\mu_b - \mu_c)^T(\Sigma_b + \Sigma_c)^{-1}(\mu_b - \mu_c)$, where μ_b and Σ_b are the mean and covariance of HSV (Hue-Saturation-Value) color of a lane-marking blob and μ_c and Σ_c are a color model's mean and covariance. We reserve an "other" class for handling all other colors of lane-marking blobs other than the two major color classes: yellow and white.

To obtain the information about the geometric structure of the current roadway, we compute an inverse perspective image from a given perspective image. The inverse perspective mapping is an image warping technique that is frequently used to remove the perspective effect from the field of lane-marking detection [1], [5], [12], [16]. This mapping essentially defines two transformations of a point, $\mathbf{X}$, from a perspective image to an inverse perspective image, $\mathbf{X}^{inv} = \mathbf{T}^{inv}_{per}\mathbf{X}^{per}$, and vice versa, $\mathbf{X}^{per} = \mathbf{T}^{per}_{inv}\mathbf{X}^{inv}$. Figure 1c shows a part of the inverse perspective image of the perspective image shown in Figure 1b.

Before we analyze the geometry of the current roadway, we further filter out false-positive lane-marking blobs from inverse perspective images where two parallel lane-markings are (nearly) parallel to each other. We removed lane-marking blobs from further consideration if their orientations were not aligned with the primary orientation. The primary orientation of lane-marking blobs is that of the longest lane-marking blob. This selection is based on the assumption that the longest lane-marking blob is always aligned with the roadway's driving direction, regardless of whether it is truly a lane-marking. For the remaining lane-marking blobs, we select any lane-marking blob pairs if their distance is probabilistically significant. In other words, we assume that the widths of legitimate road-lanes follow a Gaussian distribution, $P(w_i) \sim N(\mu, \sigma)$. We pick a lane-marking blob and a neighboring lane-marking blob. And then we compute the average distance between k selected points from the lane-marking blob pair and use that as the width between the pair. We keep the pair for further consideration if the probability of the approximated width is significant (e.g., within 1σ). This process results in a list of lane-marking blobs, some of which are in fact true longitudinal boundary lane-markings. Our approach of selecting a road-lane hypothesis is similar to that of [9] in terms of probabilistic hypothesis generation, but different in that Kim [9] used a combination of RANSAC and a particle filter to generate road-lane hypotheses.

To finalize the search of road-lane boundary lane-markings, we use the lane-marking color classification results to handpick some of the selected lane-marking blob pairs. In addition, we use two pieces of prior information: the most frequent number of road-lanes and the semantic meaning of lane-markings' colors. In particular, from government-published highway statistics [14], the majority of highways are four-lane, with two lanes each for traffic in each driving direction. In the U.S., where the vehicles drive on the right side of a road, when a driver observes a yellow lane-marking on the left side, that lane-marking almost certainly indicates the left boundary of the road-lane. This also holds true when one observes a (solid) white lane-marking on his left side to the immediate left.[1] Once we find one of lane-marking blobs on the left, either white or yellow, we choose its right-side counterpart based on the pre-defined maximum number of road-lanes. The strength of each individual hypothesis is also probabilistically evaluated as before. Figure 2 shows the results of our algorithm on recognizing the structure of a highway in Pittsburgh, PA USA. Although there are many false positive lane-marking blobs (depicted in green), the appearances of which are legitimate, our algorithm was able to pick up the right combination of lane-markings for delineating road-lane boundaries. For the internal representation, we interpolate the centerline of two identified boundary lane-markings of the host road-lane and fit a quadratic function to estimate the curvature of the current roadway.

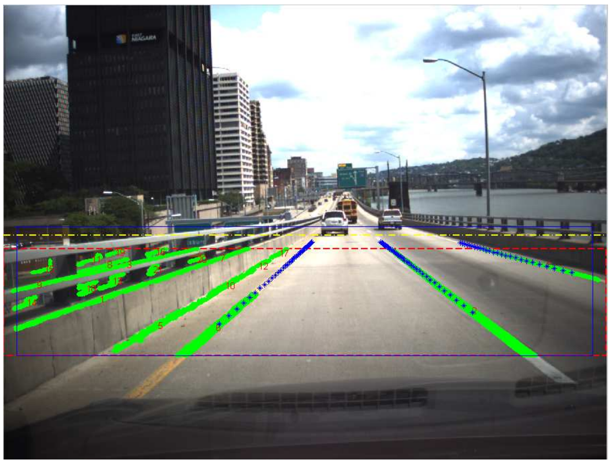


Fig. 2: The road-lane boundaries detected by our algorithm are depicted by a series of blue stars.

B. Metric Information Computation

Using the method described in the previous section, we recognized some of the detected lane-marking blobs as boundary lane-markings for the current roadway. This information enables us to understand 1) how many road-lanes are in the current roadway and 2) the index of the current road-lane from the leftmost road-lane. In the example shown in Figure 2, we know that our vehicle is driving on the leftmost road-lane of a two-lane (inter-city) highway. We now need to compute the lateral distances of our vehicle from the left and the right boundaries of the host road-lane.

To this end, we define a homography between a roadway plane and an image plane to estimate 3-dimensional coordinates of interesting points on the roadway plane. A 3D world coordinate computation through such a homography works well when the camera plane and the roadway plane are perpendicular to one another. Occasionally, however, such an assumption falls apart because of the vehicle's ego-motion and uneven ground surface. To handle with such cases, we

[1] We know which lane-marking blobs are located at the left of our vehicle because we know the image coordinates of the point our camera is projected on, in a perspective image.

estimate the angle between the camera plane and the ground plane using the vanishing point.

In what follows, we first explain how we compute a vanishing point along the horizon line and then details how we compute world coordinates of interesting points on the ground plane.

1) Vanishing Point Detection for Estimating Pitch Angle: Knowledge of a vanishing point's location and the horizon line on a perspective image provides a great deal of useful information about road scene geometry. Among these, we are interested in estimating the angle between the camera plane and the ground plane. A vanishing point is an intersection point of two parallel lines on a perspective image. In urban street scenes, one might obtain plenty of parallel line pairs, pairs such as longitudinal lane-markings and building contour lines. To obtain these contour lines and other lines, we tried three methods: Kahn's method [8], the probabilistic, and the standard Hough transform [18]. We found that the Kahn's method works best in terms of the number of resulting lines and their lengths. The Kahn's method basically uses the principal eigen vector of a pixel group's coordinates, to compute the orientation of a line fitting to that group. Figure 3 shows one result of our line extraction, where each of the extracted lines is depicted in a different color based on its orientation.

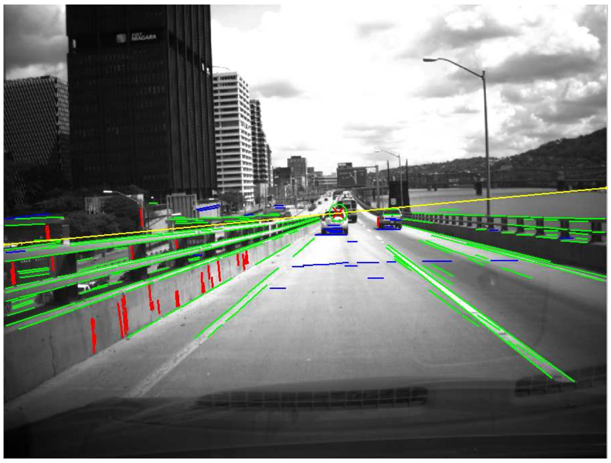


Fig. 3: An example of vanishing point detection result. The red "x" in a green circle represents the computed vanishing point along the horizon line. The yellow line represents the identified horizon line.

Given a set of extracted lines, we use RANSAC to find the best estimation of a vanishing point. In particular, we first set two priors for the horizontal and vertical line groups as $\mathbf{vp}_h = [0,0,1]^T, \mathbf{vp}_v = [0,1,0]^T$ in the camera coordinate. We then categorize each of the extracted lines into one of these two groups based on the Euclidean distance to horizontal and vertical priors. For each line pair randomly selected from the horizontal and vertical line groups, we first compute the cross-product of two lines, $\mathbf{vp}_{ij} = l_i \times l_j$, to find an intersection point. We use this intersection point as a vanishing point candidate. We then claim the vanishing point candidate with the smallest outliers as the vanishing point for that line group. A line pair is regarded as an outlier if the angle between a vanishing point candidate and the vanishing point obtained from the line pair is greater than a pre-defined threshold (e.g., 5 degrees). We repeat this procedure until a vertical vanishing point is found and more than one horizontal vanishing point is obtained. The horizon line is obtained by linking all of those horizontal vanishing points. Figure 3 shows one result of our vanishing point computation.

2) A Perspective Transformation between Camera Plane and Road Plane: This section details how we model the perspective transformation between an image plane, π, and a road plane, $\mathbf{n}$. We assume that a world coordinate frame aligned with the camera center and the roadway plane is flat. Figure 4 illustrates the perspective transformation we used in our study. The camera coordinate is oriented such that the z_c-axis is looking along a road's driving direction, the y_c-axis is looking down orthogonal to the road plane, and the x_c-axis is oriented perpendicular to the driving direction of the road. In addition, we model, based on our vehicle coordinates, the coordinate frame of the road plane such that the X_R-axis of the road plane is aligned with the z_c-axis of the camera (or world) coordinate and the Y_R-axis of the road plane is aligned with the x_c axis of the camera (or world) coordinate.

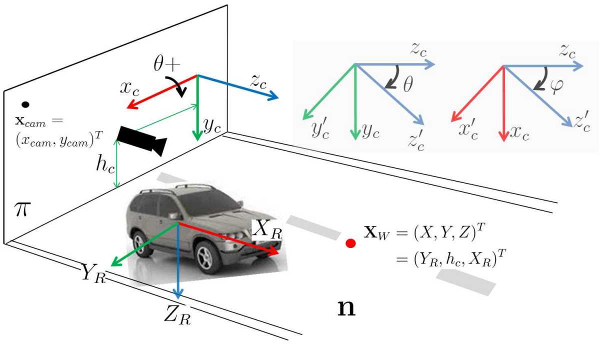


Fig. 4: A perspective transformation between the camera plane and the roadway plane.

In this setting, a point in the real-world, $\mathbf{X}_W = (X, Y, Z)$, can be represented as $\mathbf{X}_W = (Y_R, h_c, X_R)$, where h_c is the camera's mounting height from the road plane. We use the basic pinhole model [4] to define the perspective central projection between a point in the world, $\mathbf{X}_W$ and a point in a camera plane, $\mathbf{x}_{cam} = (x_{cam}, y_{cam})$. Note that a point in an image plane is further mapped throught $\mathbf{x}_{im} = K\mathbf{x}_{cam}$, where K is a camera calibration matrix defining a camera's intrinsic parameters [4].

$$\mathbf{x}_{cam} = \mathbf{P}\mathbf{X}_W \tag{1}$$

where $\mathbf{P}$ is the camera projection matrix that defines the geometric relationship between two points, $\mathbf{x}_{cam}$ and $\mathbf{X}_W$. The projection matrix, in particular, consists of a rotation matrix, $\mathbf{R}_{3\times3}(\phi, \theta, \psi)$ and a translation matrix $\mathbf{t}_{3\times1}(h_c)$, $\mathbf{P} = [\mathbf{R}(\phi, \theta, \psi)|\mathbf{t}(h_c)]$, where ϕ, θ, ψ define roll, pitch, and yaw angles. Assuming that roll and yaw angles are zero, the central projection equation is detailed as

$$\mathbf{x}_{cam} \quad = \quad [\mathbf{R}_{3\times3}|\mathbf{t}_{3\times1}] \begin{bmatrix} \mathbf{X}_W \\ 1 \end{bmatrix}_{4\times1}$$

$$\begin{aligned} &= \mathbf{R}\mathbf{X}_W + \mathbf{t} \\ \mathbf{X}_W &= \mathbf{R}^T\mathbf{x}_{cam} - \mathbf{R}^T\mathbf{t} = [\mathbf{R}^T | - \mathbf{R}^T\mathbf{t}]\mathbf{x}_{cam} \\ where\ & \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & s\theta \\ 0 & -s\theta & c\theta \end{bmatrix}, \mathbf{t} = \begin{bmatrix} 0 \\ h_c \\ 0 \end{bmatrix} \end{aligned}$$

where $c\theta$ is $\cos\theta$ and $s\theta$ is $\sin\theta$. We solve Equation 1 algebraically to obtain the coordinates of a point in the real world, (X_R, Y_R).

$$\begin{aligned} \begin{bmatrix} X_R \\ Y_R \end{bmatrix}_{2\times1} &= \begin{bmatrix} x_{cam}p_{33} - p_{13} & x_{cam}p_{31} - p_{11} \\ -y_{cam}p_{33} + p_{33} & -y_{cam}p_{31} + p_{21} \end{bmatrix}^{-1}_{2\times2} \\ &\times \begin{bmatrix} -x_{cam}(p_{32}h_c + p_{34}) + p_{12}h_c + p_{14} \\ y_{cam}(p_{32}h_c + p_{34}) - p_{22}h_c - p_{24} \end{bmatrix}_{2\times1} \end{aligned} \tag{2}$$

where (X_R, Y_R) is a point on the road plane in the world coordinate. Once we obtain these coordinates, it is straightforward to compute metric measurement of a point on the road plane. For example, X_R is the distance from the camera center.

To precisely compute such a metric measurement, it is necessary to obtain Euler angles, particularly the pitch angle, the angle between the camera plane and the ground plane. We approximate the pitch angle from a vanishing point computation in the following way. Suppose that a vanishing point at the horizon line is defined as [7]:

$$\mathbf{vp}^*_h(\phi, \theta, \psi) = \left[\frac{c\phi s\psi - s\phi s\theta c\psi}{c\theta c\psi}, \frac{-s\phi s\psi - c\phi s\theta c\psi}{c\theta c\psi}\right]^T$$

Suppose that the yaw and the roll angles are zero, the above equation yields:

$$\mathbf{vp}^*_h(\phi = 0, \theta, \psi = 0) = \left[\frac{0}{c\theta}, -\frac{s\theta}{c\theta}\right]$$

If a road plane is flat and perpendicular to an image plane, the vanishing point along the horizon line is exactly mapped to the camera center, resulting in the pitch angle being zero. From this fact, we can compute the pitch angle by analyzing the difference between the y coordinate of a vanishing point and that of the principal point, $\tan^{-1}(|P_y - vp_y|)$, where P_y is the y coordinate of the principal point.

Figure 5 presents an example result from our local roadway geometry analysis. At the top left, we display information about the geometric structure of the host roadway, such as the number of road-lanes, the index of the host road-lane, and the host road-lane's width in meters. In particular, our vehicle is driving on the first lane of a two-lane road in which the width of the host road-lane is estimated to be 3.52 meters and the true road-width is 3.6 meters. Two (red) bars along the left road-lane boundary indicate the estimated distances from the camera center (in this case, 3.80 and 9.82 meters). Finally, the lateral distances of our vehicle from the left and right boundaries are computed as 1.028 and 0.577 meters.

With this information, we can also detect whether our vehicle ever crosses a boundary of the host road-lane. In particular, we represent the estimated lateral distances of our vehicle from the left with negative numbers and from the right with positive numbers. To detect a lane-detection maneuver, we first observe these numbers up to k previous time steps (or frames), determine which lateral offset is smaller (or which side is closer to the vehicle), and claim a lane-changing maneuver when the sign of the closest side is changed. To go back to normal driving status, we observe these sequential values again and claim "normal" driving if we observe $k - l$ number of the same signs. It is important to observe a series of similar values before triggering the state change. If we only respond to a sign change, our algorithm would fail to distinguish zig-zaging from a lane-changing maneuver. Figure 6 presents a series of images as an example of lane-changing maneuver detection.

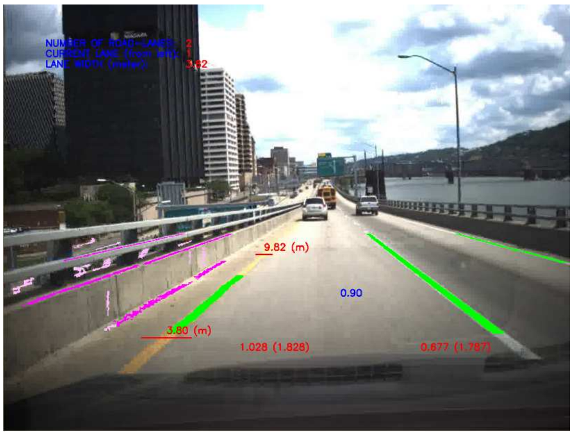


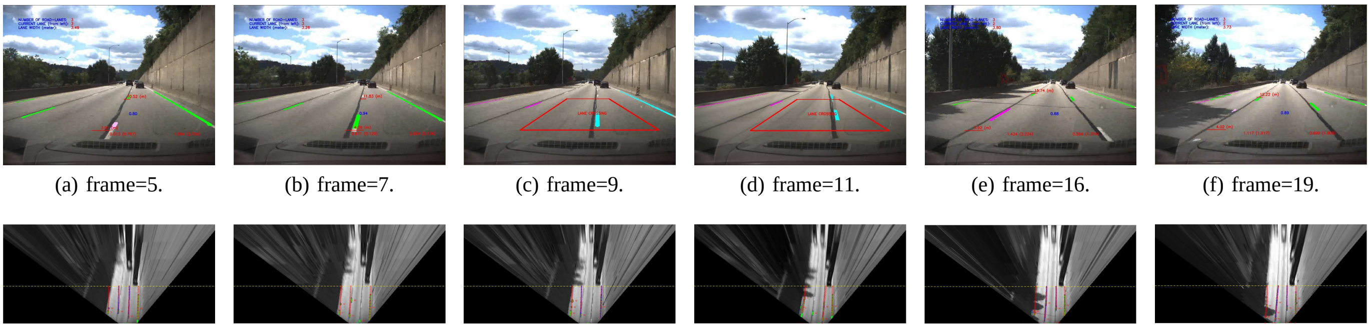
Fig. 5: An example result from our local roadway geometry analysis.

III. Experiments

In this section, we present our detailed experimental settings and results. We drove a robotic vehicle equipped with a pose estimation system. The accuracy of our pose estimator is from approximately 0.1 to 0.3 meter. We drove the vehicle one km along a curvy and hilly segment of road. Our manual measurement recorded a true lane width of 3.6 meters, but some regions of the testing path had different widths due to road geometry (i.e., intersections) or designated U-turn areas.

Figure 7 shows results of metric computation for the estimated local roadway geometry. The x-axis is time and the y-axis is computed metric in meters. A (green) dashed horizontal line is depicted at 3.6 to indicate the true lane-width. We intentionally drove the vehicle along the centerline of the testing roads until time step 400 and then, before taking a U-turn between 790 and 910, we drove the vehicle in a zig-zag fashion. While making a U-turn, our system generated no outputs, which were correct. After the U-turn, we zig-zagged at a higher fluctuation. At the upper part of the Figure, the results of lane-width computation are shown, whereas at the lower part, the results of lateral offset computation are shown, where the magenta circles (the cyan triangles) represent the left (right) lateral offsets.

On average, the lane width estimation varied between 3 and 4.5 meters with a variation of 0.342 meter. To clearly differentiate measurement errors, a different shape is depicted at the top of a lane-width estimate: A blue square for when the error is less than 0.2 meter, a cyan circle for when it is between 0.2 and 0.3 meter, and a green circle for all remaining estimates. We could improve the performance if we intentionally removed the lane-width



(a) frame=5. (b) frame=7. (c) frame=9. (d) frame=11. (e) frame=16. (f) frame=19.

(g) frame=5. (h) frame=7. (i) frame=9. (j) frame=11. (k) frame=16. (l) frame=19.

Fig. 6: An example of lane-changing maneuver detection. Images at the upper row show a series of perspective images whereas the ones at the lower row present a corresponding pairs of inverse-perspective images.

estimate, when its value is greater than 3.9 meters. Although such a thresholding is valid, in terms of using a prior information, we did not do this, to measure the accuracy as it is.

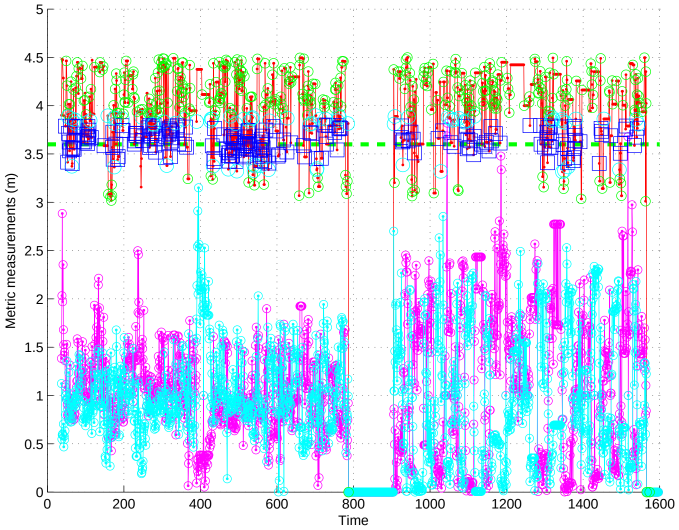


Fig. 7: Results of metric computation for the local roadway geometry. In the upper part, the results of lane-width computations are presented with different shapes based on the computation error whereas, in the lower part, the results of lateral offsets are presented in different colors (magenta (cyan) for left (right) lateral offset).

While conducting this experiment, we had no means, unfortunately, of measuring the true lateral offsets. One way we could possibly measure the performance of lateral offset computation is to look into the accuracy of the lane-width computation. This is because the error of the lateral offset computation is basically a sum of its own error and that of the lane-width computation.

To evaluate the performance of our system's lane-changing maneuver detection, we also recorded several hours of videos on different dates that included highway and city drivings. We manually identified 33 lane-changing maneuvers. We could also measure the performance of our system's metric computation from this data, but only looked at these manually identified maneuvers. For the k and l, which are the parameters for the temporal window of observing the closest lane-marking, we found 20 and 5 worked best. Our system was able to detect 27 out of 33 lane-changing maneuvers, resulting in a recall rate of (27/33 =) 0.81. Twice the system incorrectly produced outputs, resulting in a precision rate of (27/29 =) 0.93.

IV. Conclusions and Future Work

This paper has presented a computer vision system that analyzes a stream of perspective images from a forward-looking camera to acquire information about a ground vehicle's lateral movements. The outputs include the information about the geometric structure of the host roadway such as the number of road-lanes, the index of the host road-lane, and the width of host road-lane in meters. These pieces of information enabled us to determine the lateral distances of our vehicle from the left and right boundaries of the host road-lane in meters and whether our vehicle crossed any road-lane boundaries. From the actual road-tests, we found our system showed stable and reliable performance in computing lateral distance and reasonable performance in detecting lane-changing maneuvers.

As future work, we would like to investigate whether a Bayes filter would help improve the current implementation, which analyzes image frames individually in order to understand the geometric structure analysis of the host roadway. For the lane-markings' color classification, we learned, under a batch mode, the color model from a set of manually labeled color samples and used the model for the classification. The learned model is biased to the sample data and may result, when the color distribution of testing data is significantly different, in unacceptable performance. To find a remedy to this problem, we also would like to investigate whether an incremental update of the color model would help improve the performance of the color classification.

V. Acknowledgments

The author would like to thank Dr. Myung Hwangbo for his helps and fruitful discussions on 3D geometry.

References

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

[2] Nicholas Apostoloff and Alexander Zelinsky, Vision in and out of vehicles: integrated driver and road scene monitoring, *The International Journal of Robotics Research*, 23(4-5): 513-538, 2004.

[3] Itay Gat, Meny Benady, and Amnon Shashua, A monocular vision advance warning system for the automotive aftermarket, In *Proceedings of SAE 2005 World Congress and Exhibition*, 2005.

[4] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.

[5] Aharon Bar Hillel, Dan Levi, and Guy Raz, Recent progress in road and lane detection: a survey, *Machine Vision and Applications*, 2012.

[6] Albert S. Huang, David Moore, Matthew Antone, Edwin Olson, and Seth Teller, Finding multiple lanes in urban road networks with vision and lidar, *Journal of Autonomous Robots*, 26(2-3): 103-122, 2009.

[7] Myung Hwangbo, Vision-based navigation for a small fixed-wing airplane in urban environment, *Tech Report CMU-RI-TR-12-11*, PhD Thesis, The Robotics Institute, Carnegie Mellon University, 2012.

[8] P. Kahn and L. Kitchen and E.M. Riseman, A fast line finder for vision-guided robot navigation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11): 1098-1102, 1990.

[9] ZuWhan Kim, Robust lane detection and tracking in challenging scenarios, *IEEE Transactions on Intelligent Transportation Systems*, 9(1): 16-26, 2008.

[10] Jorg Kibbel, Winfried Justus, and Kay Furstenberg, Lane estimation and departure warning using multilayer laserscanner, In *Proceedings of IEEE Conference on Intelligent Transportation Systems*, pp. 777-781, 2005.

[11] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun, Map-based precision vehicle localization in urban environments, In *Proceedings of Robotics Science and Systems*, 2007.

[12] Joel C. McCall and Mohan 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.

[13] U.S. Department of Transportation, Federal Highway Administration, *Manual on uniform traffic control devices for streets and highways*, http://mutcd.fhwa.dot.gov/, 2009 Edition.

[14] U.S. Department of Transportation, Federal Highway Administration, *Highway Statistics*, http://www.fhwa.dot.gov/policy/ohim/hs06/index.htm, 2006.

[15] Michael Montemerlo et al., Junior: the Stanford entry in urban challenge, *Journal of Field Robotics: Special Issues on the 2007 DARPA Urban Challenge, Part II*, 25(9): 569-597, 2008.

[16] Marcos Nieto, Jon Arrospide Laborda, and Luis Salgado, Road environment modeling using robust perspective analysis and recursive Bayesian segmentation, *Machine Vision and Applications*, 22:927-945, 2011.

[17] Takashi Ogawa and Kiyokazu Takagi, Lane recognition using on-vehicle lidar, In *Proceedings of IEEE Intelligent Vehicle Symposiums* (IV-06), pp. 540-545, 2006.

[18] John C. Russ, *The Image Processing Handbook*, CRC Press, 2011.

[19] Sebastian Scherer, Lyle Chamberlain, and Sanjiv Singh, Autonomous landing at unprepared sites by a full-scale helicopter, *Journal of Robotics and Autonomous Systems*, 60(12): 1545-1562, 2012.

[20] Chris Urmson et al., Autonomous driving in urban environments: Boss and the Urban Challenge, *Journal of Field Robotics: Special Issues on the 2007 DARPA Urban Challenge, Part I*, 25(8): 425-466, 2008.