

Article

On the Image Sensor Processing for Lane Detection and Control in Vehicle Lane Keeping Systems

C.Y. Kuo, Y.R. Lu and S.M. Yang *

International Program on Energy Engineering, National Cheng Kung University, Tainan 70101, Taiwan; river85511@gmail.com (C.Y.K.); yoyo21308@gmail.com (Y.R.L.)

* Correspondence: smyang@mail.ncku.edu.tw

Received: 28 February 2019; Accepted: 3 April 2019; Published: 8 April 2019



Abstract: Lane keeping systems for a keeping a vehicle in the desired lane is key to advanced driving assistance system in autonomous vehicles. This paper presents a cost-effective image sensor with efficient processing algorithm for lane detection and lane control applications to autonomous delivery systems. The algorithm includes (1) lane detection by inverse perspective mapping and random sample consensus parabola fitting and (2) lane control by pure pursuit steering controller and classical proportional integral speed controller based on a nonholonomic kinematic model. The image sensor experiments conducted on a 1/10 scale model car maneuvering in a straight–curve–straight lane validate the better processing performance before, during, and after the turning section over previous work. The image sensor with the processing algorithm achieves the average lane detection error within 5% and maximum cross-track error within 9% in real-time. The development shall pave the way to cost-effective autonomous delivery systems.

Keywords: image sensor; lane detection; lane control

1. Introduction

It is known that the vast majority of all reported road accidents are due to human faults [1]. Advanced Driving Assistance System (ADAS) has recently been proposed to predict driver's intent, warn the driver about possible lane departure, and assist lane keeping [2], and it has been implemented on several vehicles in automobile industry. The reference path and the vehicle position can be determined from the maps created by geographic information system (GIS); however, the path and position accuracy often suffers from map resolution, data availability, and real-time update limitations. An on-board vision-based lane keeping system is therefore necessary to generate a reference path, obtain accurate vehicle position relative to the lane, and control the vehicle safely in a desired path.

A recent review summarized most vision-based lane detection methods that share three common steps [3]: (1) bird's eye view transformation by inverse perspective mapping [4,5], (2) lane feature detection by edge-based [6–8] or color-based methods [9], and (3) lane fitting by random sample consensus (RANSAC) combined with least-square line [10], B-spline [11], or hyperbola pair fitting [12] methods. Edge-based methods only rely on intensity information, thus require less computation for real-time applications, which proved to be key to cost-effective, autonomous vehicles in goods delivery. Lane fitting by the random sample consensus combined with parabola pair is considered the most applicable. Lane controllers can be designed based on nonholonomic kinematic or dynamic models. A proportional–integral–derivative (PID) steering controller [13] and a pure pursuit steering controller [14,15] have been proposed to determine vehicle steering angle to maintain safe driving in a lane. The above studies require on-board LIDAR (Light Detection and Ranging) with heavy computation load for lane keeping and radar for adaptive lane control.

In addition to chauffeuring people, another major function for autonomous vehicles is the delivery of goods. It is expected that 80% of all packages will be delivered autonomously in the next decade [16].

The world's e-commerce, in particular in USA and China, has doubled in the past 10 years and is expected to double again in the next five [17]. A cost-effective lane keeping system is therefore key to the development of autonomous delivery systems. This work proposes a lane keeping system for both lane detection and lane control by using only a low-cost image sensor (dashcam) with an efficient processing algorithm for real-time applications. In lane detection, inverse perspective mapping (IPM) followed by edge-based detection and RANSAC parabola fitting are applied to obtain accurate vehicle position relative to the lane center. In lane control, both pure pursuit steering controller based on a nonholonomic kinematic model and proportional-integral speed controller are adopted to maintain the vehicle safely in a desired lane. The image sensor with the processing algorithm is shown to be effective for both lane detection and lane control. The on-board processing time of the algorithm is more efficient than that of the previous work. With the advent of unmanned autonomous vehicles in the delivery market, this work may be one of the latest steps for applications to lower delivery costs of everyday items.

2. Lane Detection by Image Sensor

The vision-based lane detection is in three common steps: (1) image transformation from the sensor's frontal view to bird's eye view, (2) edge-based lane feature detection, and (3) lane markers regeneration or reconstruction in the processing image. In lane detection, the sensor image from a dashcam first has to be transformed from frontal view into bird's eye view so that the lane markings on become parallel (assuming constant lane width) for accurate vehicle positioning. Consider the image sensor with camera frame (X_c, Y_c, Z_c) mounted with pitch angle α , yaw angle θ , offset (R_x, R_y) on a vehicle in the world frame (X_w, Y_w, Z_w) at height h above the ground, as shown in Figure 1a. The transformation from an arbitrary point $P_w(x_w, y_w, -h)$ in the world frame to the corresponding point $P_i(u_i, v_i)$ in the image plane as shown in Figure 1b can be determined by coordinate transformation:

$$u_i = c_u + f_u (x_w \cos \theta - y_w \sin \theta) / d \quad (1)$$

$$v_i = c_v + f_v (-x_w \sin \alpha \sin \theta - y_w \sin \alpha \cos \theta + h \cos \alpha) / d \quad (2)$$

where (f_u, f_v) and (c_u, c_v) are the focal length and the image sensor's optical center, respectively, and $d = x_w \cos \alpha \sin \theta + y_w \cos \alpha \cos \theta + h \sin \alpha$. The inverse perspective mapping from the image plane to the ground plane can be obtained by

$$x_w = -h(-((u_i - c_u)/f_u) \cos \theta + \sin \theta(((v_i - c_v)/f_v) \sin \alpha - \cos \alpha)) / e \quad (3)$$

$$y_w = -h(((u_i - c_u)/f_u) \sin \theta + \cos \theta(((v_i - c_v)/f_v) \sin \alpha - \cos \alpha)) / e \quad (4)$$

where $e = ((v_i - c_v)/f_v) \cos \alpha + \sin \alpha$.

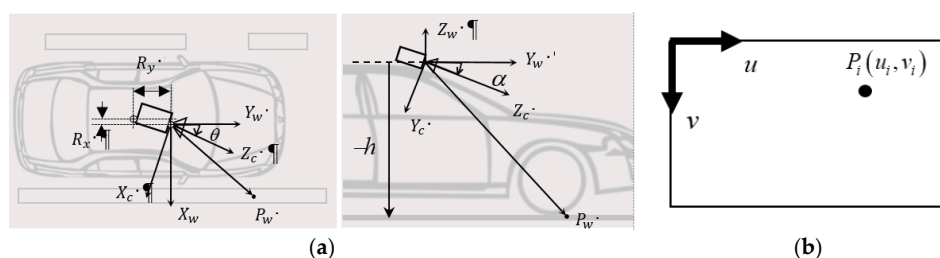


Figure 1. (a) An image sensor (dashcam) with the camera frame (X_c, Y_c, Z_c) mounted on a vehicle at height h above ground with pitch angle α , yaw angle θ , and offset (R_x, R_y) from the vehicle center in the world frame (X_w, Y_w, Z_w) , and (b) illustration of the image sensor plane, where point P_i is the projection on the image plane of a point P_w on the ground.

Figure 2a illustrates the inverse perspective mapping where the sensor image is transformed by inverse perspective mapping (IPM) into a bird's eye view image. After IPM, the lane features such as intensity and geometry of the lane markings are preserved as shown in Figure 2b, and they can be applied to locate the lane position by edge-based lane feature detection. The intensity difference between the lane markings and the ground pavement is often so strong that the IPM image can be converted to binary grayscale image and then filtered by an intensity threshold q :

$$q = kM(I) \quad (5)$$

where k is a constant to preserve lane markings, $M(I)$ is the peak with the highest intensity value in the histogram of the grayscale image, and I is the grayscale image. Figure 3a illustrates the histogram of the gray scale IPM image. The column intensity sum of the binary image can then be applied to locate the horizontal position x_L and x_R of the left and right lane marking and the lane width w , as shown in Figure 3b,c.

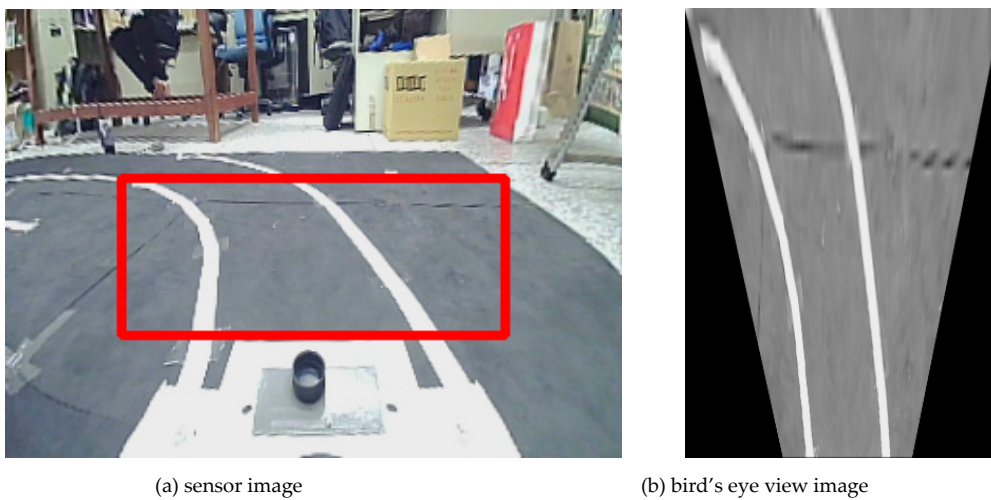


Figure 2. Illustration of the inverse perspective mapping (IPM) by converting (a) the region of interest (in box) in the sensor image to (b) the bird's eye view image for detecting the parallel lane markings.

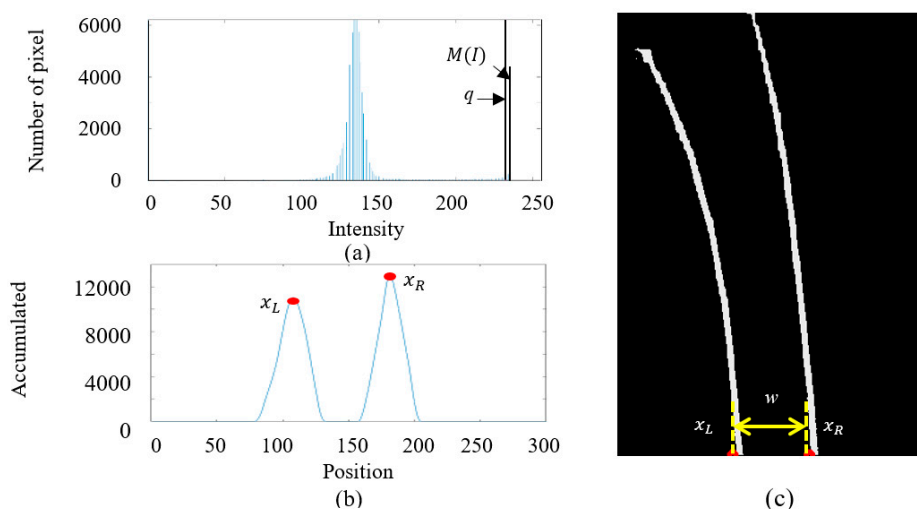


Figure 3. (a) Histogram of the grayscale bird's eye view image with $M(I)$ representing the highest intensity value and the intensity threshold q , (b) the column intensity sum of the bottom half of the image with peaks at the left and right lane markings, x_L and x_R , and (c) the result of thresholding with lane width w and position of x_L and x_R .

Note that the lane markings close to the vehicle are approximately vertical after the transformation, thus the horizontal position of the left and right lane marking can be determined by the column intensity sum of the threshold image, and two lines L_L and L_R passing through x_L and x_R , respectively, are the initial guess of the position of the left and right lane marking. Their slope can be determined by the two windows of width $w/2$ and height b at horizontal position as shown in Figure 4a. By the intensity center of each of the two windows, the slope of L_L and L_R can be determined as indicated in Figure 4b.

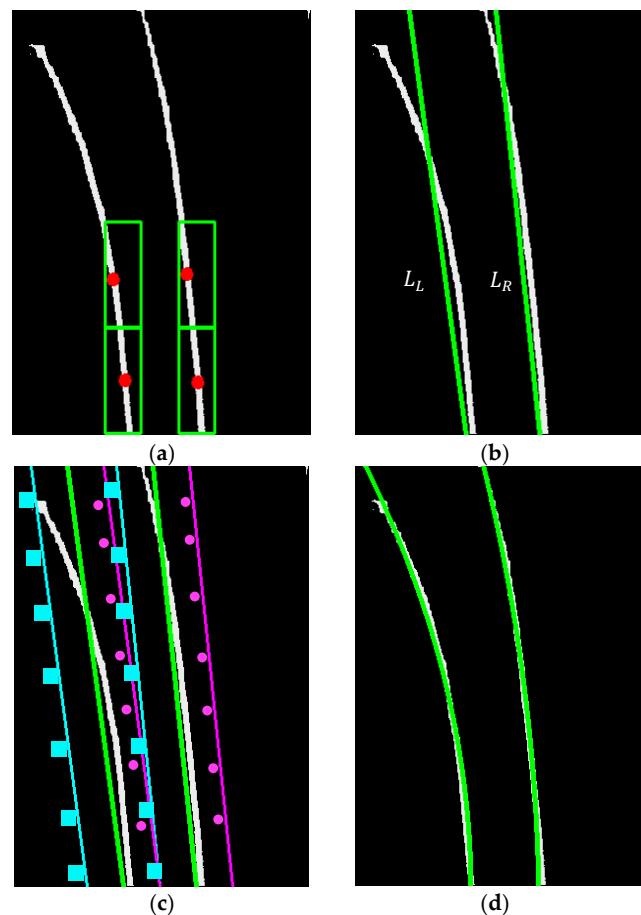


Figure 4. Illustration of (a) lane detection for the slope of the line segments L_L and L_R by the center points of the two windows at the left/right lane markings and (b) lane detection for line segments L_L and L_R from the base to the lane markings. Illustration of curve lane fitting by (c) the square lines for L_L and the dotted lines for L_R (d) by using the RANSAC parabola fitting.

One of the major challenges in autonomous delivery vehicle is lane detection when making sharp turn. In order to obtain a better position estimate in curve lane markings, random sample consensus (RANSAC) parabola fitting is applied around L_L and L_R for lane feature detection. The parabola fitting is to divide each window into sections equally spaced in the y -direction, as shown in Figure 4c, select randomly one point from each section to define parabola geometry by the least square method, and calculate the accumulated intensity value of the parabola. By use of the central limit theorem, a parabola can be estimated accurately as shown in Figure 4d. The highest accumulated intensity value in each window can be used to locate the position of each lane markings. Accurate vehicle position with respect to the lane center is necessary for lane control to minimize the cross-track error and keep the vehicle driving safely in the desired lane. The model assumes the lane markings to be parallel is applied to obtain the lane center position. The accurate vehicle position relative to the lane center can then be easily obtained by considering the offset (R_x, R_y) of the image sensor from the vehicle center.

3. Lane Control by Image Sensor

A pure pursuit steering controller and a PI speed controller were applied to keep the vehicle driving safely along the detected lane center at a desired velocity. The former controller calculates a kinematically feasible path for vehicle to maneuver from its current position to goal position. This is used for most vehicles with no universal wheels installed. The latter controller allows the vehicle to follow the calculated curve path, and classical PI control was adopted for acceptable control performance with low or no computation loading. The graphical description of a pure pursuit controller based on a nonholonomic kinematic model [18] is shown in Figure 5, where L is the vehicle wheelbase, l is the distance from the rear axle to the forward anchor point defined as the center of the vehicle, L_f is the forward drive look-ahead distance, and η is the heading of the look-ahead point (constrained to the reference path) from the forward anchor point with respect to the vehicle heading. In steering control, the steering angle δ can be determined by $\delta = \tan^{-1}(L/R)$, where R is the distance from the instantaneous rotation center O to the rear axle, $R = (L_f/2 + l \cos(\eta)) / \sin(\eta)$. The forward drive look-ahead distance L_f is dependent on the command velocity for stability. Under high velocity conditions, larger L_f is required in order to maintain the system stability and determine a more feasible δ for the nonholonomic kinematic model. In speed control, a classical PI controller was adopted without the derivative (D) term for closed loop system stability:

$$u = K_p(r - s) + K_i \int_0^t (r - s) d\tau \quad (6)$$

where u is the nondimensional speed control signal, K_p and K_i are the proportional and integral gains, respectively, r is the command velocity, and s is the vehicle velocity. The time step $d\tau \cong \Delta t$ is identified according to the update rate of the state of the vehicle and the computational speed. The proportional and integral gains K_p and K_i can be determined by extensive testing guided by the parameter space approach of robust control. By applying the parameter space approach, an area in the $K_p - K_i$ plane can be determined for which the desired design specifications such as stability, phase margin limitation, and robustness are satisfied. Through actual experiment, the PI controller parameters best fit all the design specifications are then determined.

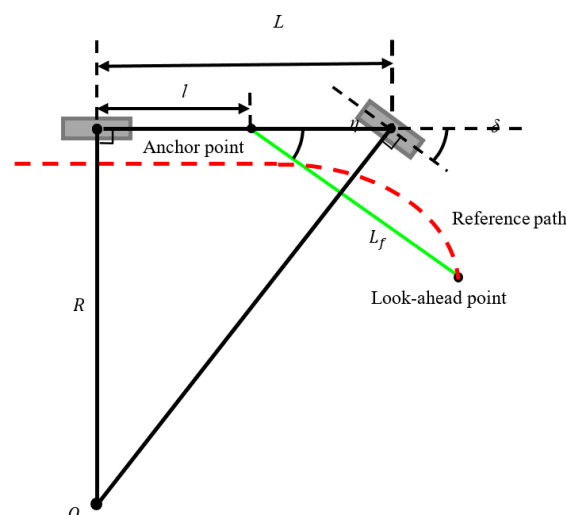


Figure 5. Illustration of lane control on a curvature of radius R with geometric description between the steering angle δ and wheel base L , the distance of the forward anchor point l , forward drive look-ahead distance L_f , and heading of the look-ahead point from the forward anchor point to the vehicle heading η .

4. Experimental Verification

The performance of the image sensor processing algorithm in lane keeping system is verified by a 1/10 scale model car of length 40 cm and width 18 cm maneuvering in straight–curve–straight lane as shown in Figure 6. The car is equipped with an image sensor (fisheye dashcam), inertial measurement unit, and on-board computer (ARM, 2 GHz). The trajectory of the car during experiment is captured by an overhead camera with 90° field of view (FOV), 24 frames per second (fps), and 800 × 600 pixel resolution to observe the cross-track error.

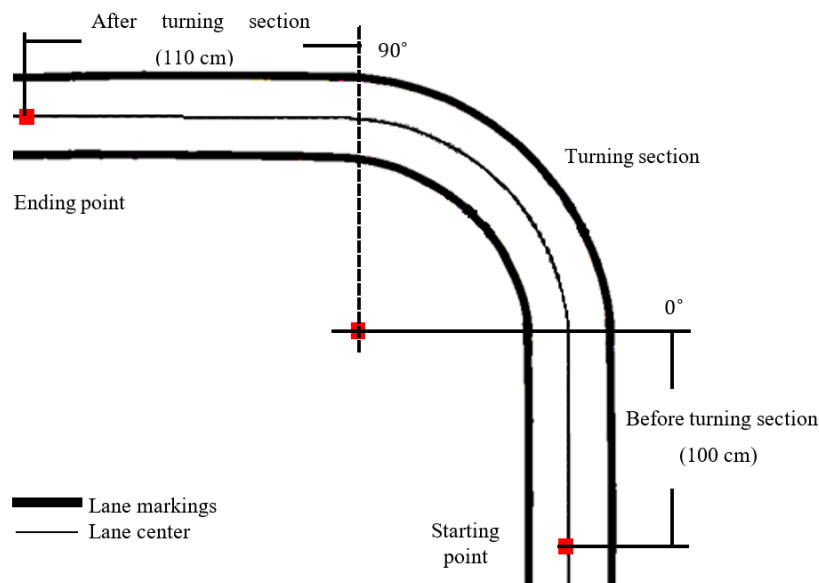


Figure 6. The route with lane center (solid line) and lane markings (thick line) for experimental verification of the image sensor in lane detection and control.

In the experiment, an image sensor with 140° field of view (FOV), 30 frames per second (fps), and 320 × 240 pixel resolution is mounted on the car at height $h = 21.3$ cm above the ground plane; yaw angle $\theta = 0^\circ$, pitch angle $\alpha = 20^\circ$, and offset $(R_x, R_y) = (0, 13.5)$ cm from the vehicle center. After camera calibration, the focal length (f_u, f_v) is (189.926, 256.917) pixels, while the camera optical center (c_u, c_v) is (160.717, 120.688) pixels. Most lane markings belong to a region of interest (ROI) of 260 pixel width and 85 pixel height with the position of the top left corner (30, 90) in the image sensor, as shown in Figure 2a. The image in ROI is then transformed to the bird's eye view image (300 × 400 pixel) and converted to grayscale as shown in Figure 2b. The intensity histogram of the grayscale IPM image is calculated with the peak marking the highest intensity $M(I)$ as shown in Figure 3a. According to the testing results, $k = 0.9$ is adopted to obtain the best filtering performance by Equation (5). The column intensity sum of the binary image can then be applied to locate the horizontal position x_L and x_R of the left and right lane marking and the lane width w as shown in Figure 3b,c. With the two windows of width $w/2$ and height 60 pixels at the bottom of each of the two lane markings as shown in Figure 4a, the slope of each lane marking can be determined as shown in Figure 4b. These slopes are applied to obtain two windows of equal width for spline fitting as shown in Figure 4c. Each window is divided into 10 sections and the number of iterations of the spline fitting is set of 30 to obtain the best performance on the on-board computer, as shown in Figure 4d. Based on the parallel lane model, the lane center can therefore be obtained.

The efficiency of the lane detection algorithm is shown in Figure 7a. In a series of 300 captured images of a typical maneuver in a straight lane, the computation time in lane detection by the on-board CPU (2 GHz) is in the range of 10.4 to 11.2 ms. By comparison, the computation time by using the method in [6] is 11.5 to 12.2 ms. The algorithm in this work is shown more efficient in all of the 300 images in lane detection. In addition, the algorithm remains applicable to curve lane, while the

method in [6] is otherwise for its limitation when using Hough transform. Figure 7b illustrates the accumulated time of processing the 300 images during vehicle motion of ~10 s. The algorithm is shown to shave 0.5 s of computation time in lane detection, and such saving is critical to the development of autonomous delivery systems.

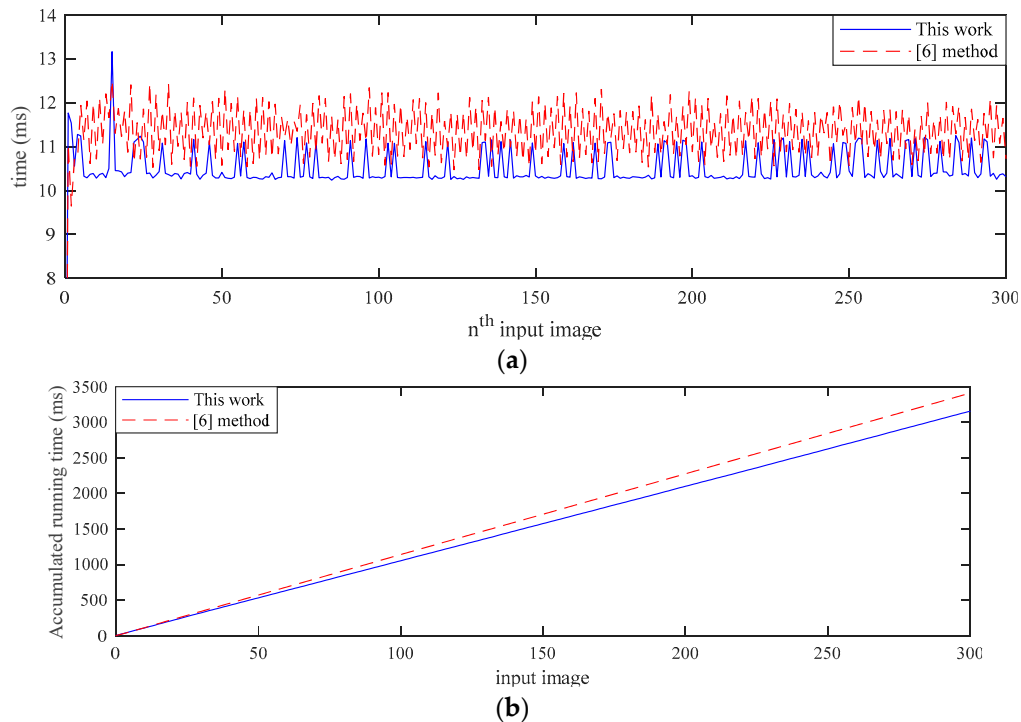


Figure 7. (a) Comparison of the processing time in lane detection for a series of 300 images and (b) the accumulated computation time validates that the algorithm in this work is more efficient.

The lane control experiment is also performed on the 1/10 scale model car. For the pure pursuit controller, the vehicle wheel base is $L = 26$ cm and the distance from the rear axle to the forward anchor point is $l = 6$ cm. The forward drive look-ahead distance L_f is dependent upon the command velocity r to overcome the stability issue. According to the characteristics of the nonholonomic kinematic model, the relationship between L_f and r in the experiment is

$$L_f(r) = \begin{cases} 55 \text{ cm} & \text{if } r < 1.35 \text{ (m/s)} \\ 43 \text{ cm} & \text{if } 1.35 \text{ (m/s)} < r < 1.5 \text{ (m/s)} \\ 65 \text{ cm} & \text{otherwise.} \end{cases}$$

For speed control, the PI controller parameters K_p and K_i are determined through the parameter space by experiment: $K_p = 0.3$ and $K_i = 0.04$. For the integration, the time step is set at 5 ms to achieve the best performance.

The experiments were conducted with the model car maneuvering at a speed of 1 m/s in a straight–curve–straight lane (radius 99 cm) and lane width 37 cm. The vehicle speed is set at high enough to generate ~1G centrifugal acceleration during the turning section to simulate real-world vehicle operation. The experiment results for lane detection and lane control are shown in Figure 8, where the solid line represents the real lane center, the dotted line the lane control results, and the dash line the lane detection result. The error percentage is defined by the ratio of deviation over the lane width. Since the width of the lane markings in the experiment is 35 cm, the lane detection error is expected to be within ± 1.75 cm ($\pm 5\%$). As for cross-track error, the maximum tolerable cross-track error is ± 9.5 cm ($\pm 25\%$), which is the maximum offset of the vehicle from the lane center without crossing the lane markings. The average lane detection error and maximum cross-track error in these

three sections are 2.54%, 3.37%, 4.41%, and -4.44% , -8.89% , -4.08% , respectively. Due to the small steering angle ($3^\circ - 5^\circ$) adjustment is not possible, and there is a slight increase during the turning section. Even if the lane keeping system is capable of detecting the small offset from the lane center and sending a corresponding command to the steering angle during the turning section, the car is still not capable of adjusting its steering angle due to backlash in hardware constraint. However, even under this limitation, the performance of the vision-based lane keeping system can still be clearly indicated by the maximum cross-track error in the experiment within 9%.

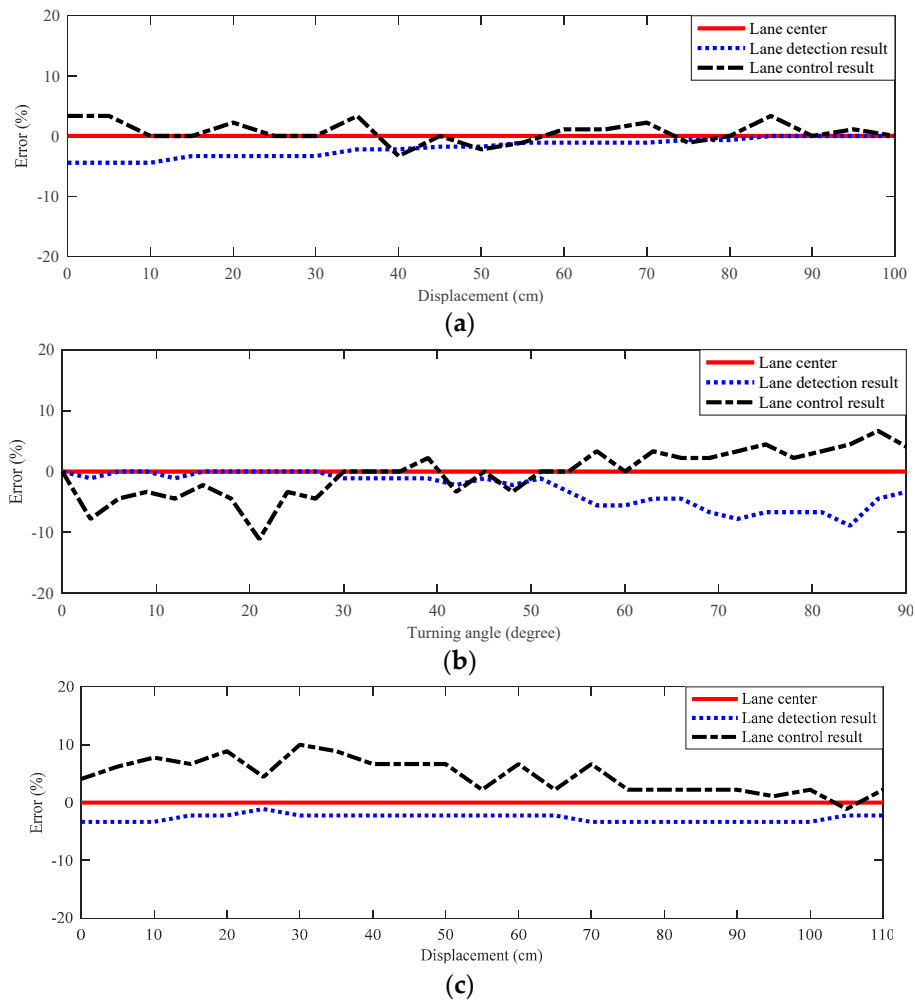


Figure 8. Error percentage in the experimental results of the image sensor in lane keeping system (a) before, (b) during, and (c) after the turning section, where the error percentage is defined as the ratio of the offset from the lane center to the lane width.

5. Conclusions

- (1) A cost-effective lane keeping system is key to the development of autonomous delivery systems. This work proposes a lane keeping system for both lane detection and lane control by using only a low-cost image sensor (dashcam) with an efficient processing algorithm for real-time applications. A vehicle with only an image sensor, without LIDAR or radar, is shown to be capable of lane detection and lane control with good accuracy in real-time. The image processing algorithm in lane detection includes (a) inverse perspective mapping to transform the dashcam image to bird's eye view image, (b) binary and histogram filter to detect the lane slope feature, and (c) random sample consensus parabola fitting to reconstruct the lane markings.
- (2) The performance of the image sensor processing algorithm in lane keeping system is verified by a 1/10 scale model car of length 40 cm and width 18 with an image sensor (fisheye dashcam),

inertial measurement unit, and on-board computer (ARM, 2GHz). The efficiency of the lane detection algorithm is ~11 ms for lane detection of a typical image, which is ~5–10% lower than that in comparable study [6]. The algorithm in this work is shown more efficient in all of the 300 images in lane detection. In addition, the algorithm remains applicable to curve lane. In lane control, the trajectory of the car during experiment is captured by an overhead camera to observe the cross-track error. Experiments conducted on the model car maneuvering at speed 1 m/s in a curve lane of radius 99 cm for ~G acceleration further show that that the average lane detection error before, during, and after curve lane are all within 5%. The image sensor with the processing algorithm is effective in advanced driving assistance of autonomous delivery systems. In lane control, a pure pursuit steering controller and a proportional integral speed controller based on a nonholonomic kinematic model are applied to show that the maximum cross-track error is within 9% in lane control. Both the lane detection and lane control performance are critical to the development of autonomous delivery systems.

- (3) The proposed lane keeping system is considered effective and efficient. Future development will simulate more complicated road conditions such as rugged ground and light conditions, such as glare when facing the sun and dim light when driving in a tunnel. Rugged ground may cause the vehicle to tilt, thus changing the yaw and pitch angle of the vehicle, and affecting the accuracy of inverse perspective mapping. Glare and dim light conditions may reduce the intensity difference between the lane markings and the road pavement, so that detecting lane features using the edge-based method may therefore be challenging.

Author Contributions: C.Y.K. and Y.R.L. conducted the experimental verifications in their joined capstone project, and S.M.Y. provided the algorithm development and oversaw the project progress.

Funding: Kuo and Lu are grateful to the summer research grants from the Ministry of Science and Technology of Taiwan government.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *Global Status Report on Road Safety 2015*; World Health Organization: Geneva, Switzerland, 2015.
2. Skog, I.; Handel, P. In-Car Positioning and Navigation Technologies: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 4–21. [[CrossRef](#)]
3. Hillel, A.B.; Lerner, R.; Levi, D.; Raz, G. Recent Progress in Road and Lane Detection: A Survey. *Mach. Vis. Appl.* **2014**, *25*, 727–745. [[CrossRef](#)]
4. Hannan, M.; Javadi, M.; Samad, S.; Hussain, A. A Robust Vision-based Lane Boundaries Detection Approach for Intelligent Vehicles. *Inf. Technol. J.* **2012**, *11*, 1184–1192. [[CrossRef](#)]
5. Ding, Y.; Xu, Z.; Zhang, Y.; Sun, K. Fast Lane Detection Based on Bird's Eye View and Improved Random Sample Consensus Algorithm. *Multimed. Tools Appl.* **2016**, *76*, 1–20. [[CrossRef](#)]
6. Low, C.Y.; Zamzuri, H.; Mazlan, S.A. Simple Robust Road Lane Detection Algorithm. In Proceedings of the IEEE 2014 5th International Conference on Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, 3–5 June 2014.
7. Mammeri, A.; Boukerche, A.; Tang, Z. A Real-time Lane Marking Localization, Tracking and Communication System. *Comput. Commun.* **2016**, *73*, 132–143. [[CrossRef](#)]
8. Niu, J.; Lu, J.; Xu, M.; Lv, P.; Zhao, X. Robust Lane Detection Using Two-stage Feature Extraction with Curve Fitting. *Pattern Recognit.* **2016**, *59*, 225–233. [[CrossRef](#)]
9. Rotaru, C.; Graf, T.; Zhang, J. Color Image Segmentation in HSI Space for Automotive Applications. *J. Real-Time Image Process.* **2008**, *3*, 311–322. [[CrossRef](#)]
10. Son, J.; Yoo, H.; Kim, S.; Sohn, K. Real-time Illumination Invariant Lane Detection for Lane Departure Warning System. *Expert Syst. Appl.* **2015**, *42*, 1816–1824. [[CrossRef](#)]
11. Lim, K.H.; Seng, K.P.; Ang, L.-M. River Flow Lane Detection and Kalman Filtering-Based B-Spline Lane Tracking. *Int. J. Veh. Technol.* **2012**, *2012*, 465819. [[CrossRef](#)]

12. Du, X.; Tan, K.K. Vision-based Approach towards Lane Line Detection and Vehicle Localization. *Mach. Vis. Appl.* **2015**, *27*, 175–191. [[CrossRef](#)]
13. Marino, R.; Scalzi, S.; Netto, M. Nested PID Steering Control for Lane Keeping in Autonomous Vehicles. *Control Eng. Pract.* **2011**, *19*, 1459–1467. [[CrossRef](#)]
14. Samuel, M.; Hussein, M.; Binti, M. A Review of Some Pure-Pursuit Based Path Tracking Techniques for Control of Autonomous Vehicle. *Int. J. Comput. Appl.* **2016**, *135*, 35–38. [[CrossRef](#)]
15. Shan, Y.; Yang, W.; Chen, C.; Zhou, J.; Zheng, L.; Li, B. CF-Pursuit: A Pursuit Method with a Clothoid Fitting and a Fuzzy Controller for Autonomous Vehicles. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 1–13. [[CrossRef](#)]
16. *Autonomy Delivers: An Oncoming Revolution in the Movement of Goods*; KPMG: New York, NY, USA, 2018.
17. *Why Self-Driving Vehicles Are Going to Deliver Pizzas before People*; Fortune: Somerset, NJ, USA, 2018.
18. Kuwata, Y.; Karaman, S.; Teo, J.; Frazzoli, E.; How, J.; Fiore, G. Real-time Motion Planning with Applications to Autonomous Urban Driving. *IEEE Trans. Control Syst. Technol.* **2009**, *17*, 1105–1118. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).