

Article

Lane Detection Algorithm Using LRF for Autonomous Navigation of Mobile Robot

Jong-Ho Han ¹ and Hyun-Woo Kim ^{2,*}

¹ Test & Evaluation Division, Korea Intelligent Automotive Parts Promotion Institute, 201, Gukgasandanseo-ro, Guji-myeon, Daseong-gun, Daegu 43011, Korea; jonghohan@kiapi.or.kr

² Marine IT Convergence Materials Research Division, Research Institute of Medium & Small Shipbuilding, 38-6, Noksan industrial Complex 232-ro, Gangseo-gu, Busan 46757, Korea

* Correspondence: kimhw@rims.re.kr; Tel.: +82-051-974-5549

Abstract: This paper proposes a lane detection algorithm using a laser range finder (LRF) for the autonomous navigation of a mobile robot. There are many technologies for ensuring the safety of vehicles, such as airbags, ABS, and EPS. Further, lane detection is a fundamental requirement for an automobile system that utilizes the external environment information of automobiles. Representative methods of lane recognition are vision-based and LRF-based systems. In the case of a vision-based system, the recognition of the environment of a three-dimensional space becomes excellent only in good conditions for capturing images. However, there are so many unexpected barriers, such as bad illumination, occlusions, vibrations, and thick fog, that the vision-based method cannot be used for satisfying the abovementioned fundamental requirement. In this paper, a three-dimensional lane detection algorithm using LRF that is very robust against illumination is proposed. For the three-dimensional lane detection, the laser reflection difference between the asphalt and the lane according to color and distance has been utilized with the extraction of feature points. Further, a stable tracking algorithm is introduced empirically in this research. The performance of the proposed algorithm of lane detection and tracking has been experimentally verified.



Citation: Han, J.-H.; Kim, H.-W. Lane Detection Algorithm Using LRF for Autonomous Navigation of Mobile Robot. *Appl. Sci.* **2021**, *11*, 6229. <https://doi.org/10.3390/app11136229>

Academic Editor: Dario Richiedei

Received: 20 April 2021
Accepted: 2 July 2021
Published: 5 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

There has been considerable progress in the field of vehicle safety systems in the last few decades—from safety belts in the 60s to electrical systems such as air bags, anti-lock brake systems (ABSs), and electric power steering (EPS) in the 90s; this progress has been aimed at increasing passenger safety. However, these systems offer only passive safety because they operate in response to the state of the in-vehicle system [1]. In line with the recent development of smart technologies, considerable research has been conducted on the implementation of intelligent and sophisticated safety systems [2]. Such a system is referred to as an advanced safety vehicle (ASV). An ASV is a safety system designed to alarm drivers in advance against accident-prone situations so that they can cope with such situations actively. ASV-enabled vehicles are cutting-edge, safe vehicles equipped with artificial intelligence involving various types of intelligent safety technologies and thus maximizing driving safety and comfort. ASV's representative core technologies are a vehicle collision alarm system and a lane recognition system. The former is a vehicle-to-vehicle distance detection sensor that alarms the driver when the distance from the vehicle in front or behind becomes alarmingly small. The latter, commonly called a lane departure warning system (LDWS), is a sensor detecting the vehicle leaving its lane; it checks the lane and informs the driver of an imminent lane departure with an audible signal. The sensors mainly used in ASV are an ultrasonic sensor to recognize obstacles in front, a vision-based system to recognize the lane and road situations such as traffic lights with radar, and a laser range finder (LRF). The information detected is used for

recognizing obstacles, map building, and driving, thereby ensuring location detection, collision avoidance, path tracking, lane recognition, etc. [3–6]. Lane recognition technology, one of the core technologies of ASV, involves a vision-based system and a laser-sensor-based system, among others. The vision-based system has a large capacity of extracting road information with low-cost equipment, but its drawback is its difficult processing due to its oversensitivity to illumination conditions, which makes it easily saturated when there is a shadow or a change in luminosity. Furthermore, we tested the performance of a vision-based lane recognition system in a previous study by varying the techniques, such as Hough transform, template matching, splines, and polynomial estimation, depending on the road conditions, but it showed vulnerability to some curved road conditions [7,8]. In order to overcome this drawback, research studies have been conducted using various types of road modeling, taking into account road properties such as horizontal and vertical clothoid curves [9], polynomial curves [10,11], and splines [12], but these methods are sensitive to noise or false detection and require an elaborate control method using filters or the like to remove noises or reduce recognition errors [13]. Further, the number of frames should be maintained high in order to ensure a stable recognition. Such elaborate control and frame number maintenance are considerable cost factors, and because of this method's sensitivity to weather conditions, particularly under foggy conditions that reduce visibility, difficulties are encountered while judging road conditions and recognizing lanes. Unlike the vision-based system, an LRF-based system is not influenced by illumination changes. It can recognize lanes even in the dark because it measures distance using the returning emitted laser signal. Moreover, it responds only to specific frequencies such that steep upward driving does not affect it. It can easily recognize lanes even under foggy weather conditions. These advantages keep attracting research interests to LRF-based lane recognition [14–16]. Previous studies fused the vision-based system or global positioning system (GPS) data with LRF data and used LRF as an auxiliary tool for lane recognition or built 2D maps utilizing the reflectance of LRF [17–19]. When LRF is used as an auxiliary tool, it is used for determining the position of a lane by comparing it with that recognized by a vision-based system at a measurement point or within its scope, or simply for detecting the location of a curb. In such cases, different sensors should be fused to build a system, which is inevitably expensive. When a 2D map is built using a single LRF, noises should be removed to facilitate the lane recognition, which requires the use of various complicated filters. Moreover, it cannot easily discern obstacles [20–26]. On the other hand, when a 3D map is built using a single LRF, the criteria for lane recognition becomes clear and the obstacles can be detected more easily. Noise removal for lane recognition becomes easier as well by simply deleting the parts other than the lane-forming height.

In this study, road conditions were scanned using a single LRF, and a 3D road map was built. Further, lanes were extracted from the map built by means of the feature point extraction algorithm so that they could be recognized. Typically, road lanes are divided by continuous lines and dashed lines. Continuous lines can be easily recognized and tracked with a 3D road map and the feature point extraction algorithm. However, when these lines become faded or broken over time, lane recognition becomes difficult and accidents are more likely to occur. To prevent such accidents, we present in this paper a lane prediction method using a curvature algorithm. This algorithm has the advantages of simple calculation and excellent expression of simple curves. It is, therefore, suitable for rapid processing.

The proposed method was designed to enable a lane tracking system to track a discontinuous lane by predicting the path of the lane lying ahead based on the lane tendencies figured out from those recognized earlier and by applying the curvature theory. In the case of dashed lines, precise path prediction was implemented by updating the coordinates of the dashed-line lanes along the path predicted with the curvature algorithm. This method can be effective not only in reducing the real-time system load compared to the lane prediction by means of a complicated prediction algorithm, but also in allocating system resources. The rest of this paper is organized as follows: Section 2 explains the

structure and modeling of the system used in this study. Section 3 explains the algorithm with which a 3D road map is built with 2D data and the method for extracting feature points. Section 4 explains how lane tendencies can be determined for the prediction of a discontinuous lane and the prediction method using the curvature algorithm. Section 5 explains a stable tracking algorithm for implementing tracking by a mobile robot by recognizing lane feature points acquired from the 3D road map. Finally, in Section 6, the experimental results and conclusions of this study are presented.

2. System Structure

Figure 1 shows the configuration of the mobile robot system. This system is mainly divided into a mobile robot platform and a data control platform. The upper part of the dashed line represents the mobile platform, and the lower part, the data control platform. The mobile platform is subdivided into sensor and motor controller units, on the left and right sides of Figure 1, respectively.

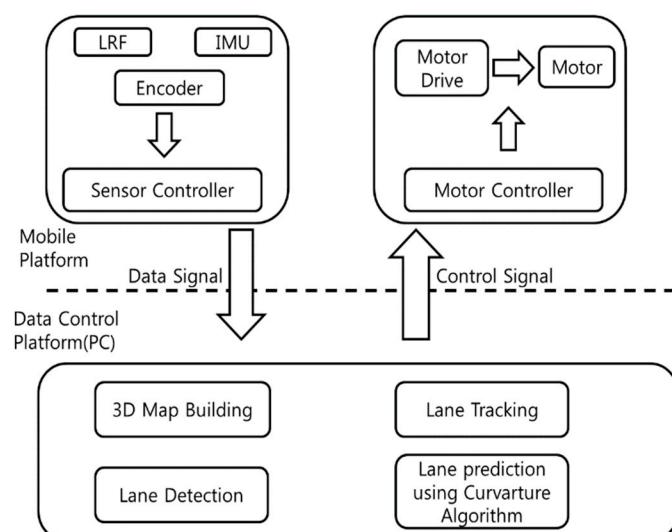


Figure 1. System configuration.

The sensor unit of the mobile platform is composed of the LRF, inertial measurement unit (IMU), and encoder as well as the sensor controller. Sensors measure the information necessary for building road maps and controlling the robot, and the data controller receives the data values obtained by the sensors and ensures a stable transfer to the data control platform. LRF is used for obtaining the road information by scanning the road using a laser. The IMU sensor is used for obtaining information on the position and direction of the robot. The encoder is used for measuring the rotational values of the mobile robot's motor and obtaining the information on distance covered and relative coordinates by means of dead reckoning. The motor control unit of the mobile platform is composed of the motor drive, motor, and motor controller. The motor controller interprets the control signals received from the data control platform in order for the mobile robot to navigate stably and the motor drive for moving the mobile robot according to the instructions from the motor controller. The data control platform consists of a personal computer (PC)-based program and executes functions such as creation of 3D road maps with the data transferred from the mobile platform sensor unit, extraction of lane feature points, prediction of lanes, and generation of control signals. Further, it transfers the generated control signals to the mobile platform motor control unit for the mobile robot to navigate stably along the lane. Figure 2 shows the structure of the robot built for this study. The LRF on the right-hand side is a sensor used for obtaining road information. The sensors produced by Hokuyo Ltd. and Sick AG are mostly widely used, and the URG-04LX of the Hokuyo Ltd. used in this study has a relatively narrow measurement range compared to the sensor by Sick AG, but

its minimum measurement distance is very close and its resolution makes it suitable for building 3D maps and scanning lanes.

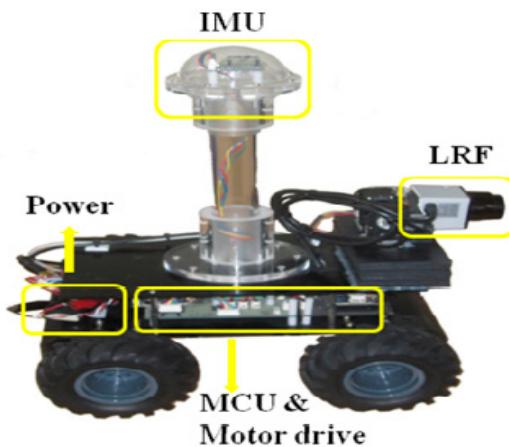


Figure 2. Structure of the mobile robot.

The IMU positioned at the upper end is a module composed of an accelerator, gyro, and compass sensors, and is used for measuring the direction and position of the robot. Since it is considerably influenced by iron or a magnet, a sufficient distance should separate it from the robot's main body, motor, and the other parts. The micro controller unit (MCU) placed at the bottom part is a sensor controller sending data to the PC and a motor controller that transfers the instructions received from the PC to the motor drive. These two controllers and the motor drive are positioned at the center. The components used in the experiment are listed in Tables 1–4.

Table 1. Laser range finder: UBG-04LX.

	Light Source	Semiconductor laser diode (785 nm) Laser safety Class1 (IEC60825-1)
	Detection Distance Standard Object	Accuracy Range: 60~4095 mm Square Kent Sheet 80 mm
UBG-04LX	Detection Distance /Scan Angle	20~4000 mm/240°
	Resolution/ Angular Resolution	1 mm/0.36°(360° / 1024 steps)
	Scan Time	100 msec/scan
	Interface	RS-232C (19.2, 57.6, 115.2, 500, 750 kbps) USB 2.0 (Full Speed)

Table 2. The specifications of IMU (Inertial Measurement Unit).

IMU (EBINU-9DOF)	Input Voltage	5 V
	Current	40 mA
	Data Output Velocity	1~100 Hz
	Sensitivity	Gyro: 250~2000 dps Acceleration: 2~8 g Compasses: 1.3~8.1 gauss
	Resolution	0.01 degree
	Size (L, W)	15 mm, 23.5 mm
	Interfaces	TTL Level (3.3 V) Serial Communication

Table 3. The specifications of motor and motor driver.

Motor (RB35GM)	Rated Voltage	12 V
	Rated Torque	2.0 g-cm
	Rated Speed	102 rpm
	Output Power	3.14 W
	Deceleration	50:1
Motor Driver (NT-S-DCDM1210)	Input Voltage Range	12~24V
	Maximum Continuous Current	5 A
	PWM Frequency	14.4 Khz
	Interface	RS-232, I2C

Table 4. Distance and angle measurements.

Controller (Mycortex-LM8962)	Power	3.3 V
	Clock	50 Mhz
	Memory	256 kB flash/64 kB SRAM
Product Features	32-Bit RISC, GPT, SSI, UART, ADC, I2C, PWM, GPIO	

3. Principles of LRF and 3D Map Building

3.1. Principle of LRF Lane Scanning

LRF measurement methods can be classified by time of flight measurement (TOF), triangular measurement, and phase-shift comparison measurement [27]. In this study, the phase-shift comparison measurement was used. Its advantages are a precise measurement at a very short distance and a low error range. This method involves electrical signals emitted by reflecting the phase and intensity of the laser, whereby the sensor measures the distance by comparing the intensities of the initially emitted signals and the returned signals reflected off the target using Equation (1).

$$L = \frac{1}{2} \times \theta \div \left(\frac{f}{c} \times 2\pi \right) \quad (1)$$

where L denotes the distance to the target; θ , the phase shift; f , the laser frequency; and c , the velocity of light [15].

When a sensor is used in field environments to measure the phase shift, the intensity of the laser returning to the sensor reflected off the target object decreases because of external factors such as the distance of the target, angle, and optical reflection rate of the target [28]. Because diminished laser signals make the phase difference detection more difficult, the signals should be amplified before being interpreted; however, this process is prone to calculation errors.

Figure 3 shows a schematic representation of the laser signal processing steps. The returning light reflected off the object is converted into electrical signals at the avalanche photodiode (APD) and amplified with an automatic gain control (AGC) amplifier. In this process, the closer the color of the object is to black, the less is the amount of energy returned, resulting in weak signals as compared to objects having other colors; thus, the signals are amplified at AGC and calculated to be located closer than other objects [29]. When this principle is applied to road scanning, a height difference in the asphalt lane is observed.

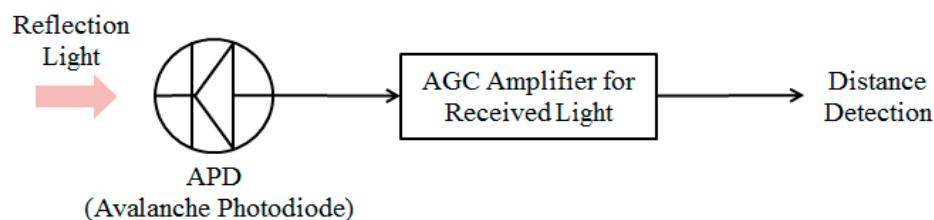


Figure 3. Data processing of the LRF.

Figure 4 shows the LRF-based road scan principle. As stated above, a black-colored object scanned with LRF is considerably amplified by AGC and the LRF-to-object distance appears to be shorter than it really is, whereas colored objects appear almost identical to their real distances [30,31]. As a result, in a 3D road map built using the LRF data, asphalt seems to be higher than the lane.

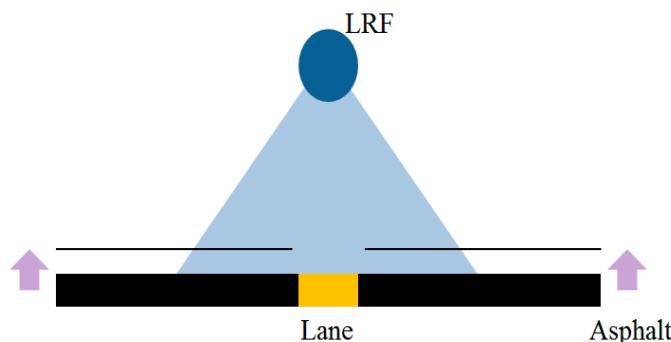


Figure 4. Road scan of LRF.

3.2. 3D Map Building

The LRF sensor used by the mobile robot in this study transmits the distance and angle data to the data control platform, which generates x and y coordinates using the angle and distance data as expressed by Equations (2a) and (2b) and builds a 2D map [32,33].

$$x_{LRF} = D \times \cos(\theta_{LRF}) \quad (2a)$$

$$y_{LRF} = D \times \sin(\theta_{LRF}) \quad (2b)$$

where D denotes the LRF-to-object distance and θ_{LRF} represents the angle of the LRF scan. LRF can scan a range of 1–240° and scan the surrounding area sequentially rotating by 0.36°.

The data coordinated by LRF do not match the coordinate axis of the mobile robot. Thus, to overcome this problem, coordinate transformation is necessary to match the standard coordinate axis of the mobile robot. Figure 5 shows the coordinate transformation process in which the LRF 2D scan data undergo transformation to match the coordinate axis of the mobile robot. X0, Y0, and Z0 constitute the standard coordinates used for building maps with the coordinate axis of the mobile robot. X1, Y1, and Z1 are the LRF coordinates. The change in the coordinates of the mobile robot and position entails a change in the LRF's coordinate axis; in order to build a 3D map using the LRF-measured data, LRF data should be coordinated to match X0, Y0, and Z0. This requires the movement, rotation, and LRF coordinate axis data; further, Equation (3) using the rotation matrix is applied to implement the transformation of the LRF coordinates.

$$\begin{bmatrix} x_{map} \\ y_{map} \\ z_{map} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_{Robot} \\ y_{Robot} \\ z_{Robot} \end{bmatrix} + \begin{bmatrix} x_{LRF} \\ 0 \\ y_{LRF} \end{bmatrix} \quad (3)$$

where x_{Robot} , y_{Robot} , and z_{Robot} denote the mobile robot's coordinates; x_{LRF} , y_{LRF} represent the coordinates of the data scanned by LRF; θ denotes the rotational angle of the mobile robot's coordinates around X_0 ; and x_{map} , y_{map} , z_{map} represent the coordinates of the points for building a map. The coordinates of the LRF data transformed with Equation (3) are used for building a 3D map through a continuous iteration.

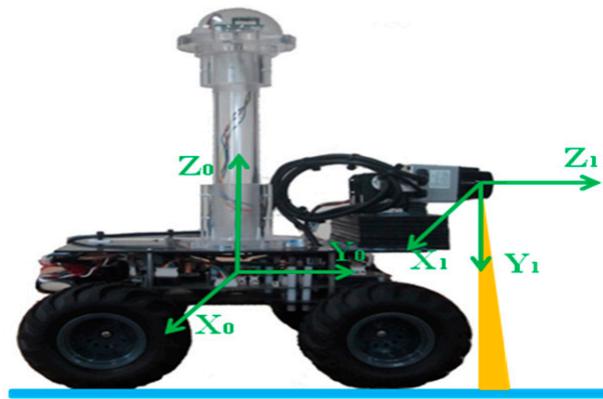


Figure 5. Coordinate transformation of map data.

Figure 6 shows a 3D road map exhibiting the principles shown in Figures 4 and 5. This 3D road map represents the LRF-scanned test environment, wherein the red circle denotes a lane. The inset shows the test environment.

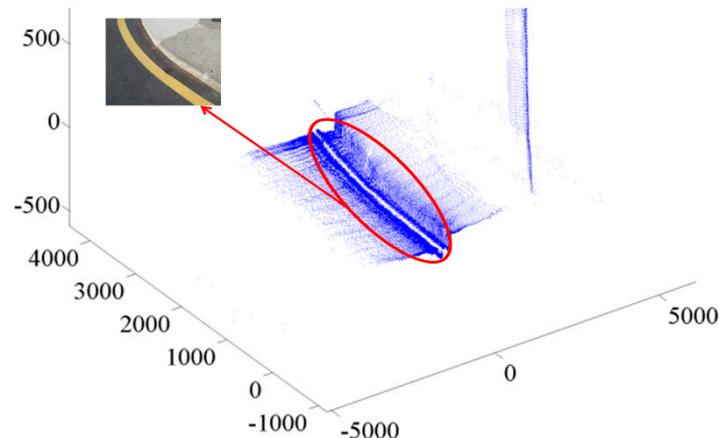


Figure 6. The 3D road map.

4. Lane Recognition and Prediction

4.1. Lane Feature Point Extraction

In order to discern the line between lane and asphalt, it is necessary to extract the edge points of the lane formed lower than the road with respect to the road surface. We have used Equations (4a) and (4b) for extracting these boundary points.

$$\sqrt{(z_{map(i)} - z_{map(i+1)})^2} \geq d \quad (4a)$$

$$\left| \tan^{-1} \left(\frac{z_{map(i)} - z_{map(i+1)}}{x_{map(i)} - x_{map(i+1)}} \right) \right| \geq \sigma \quad (4b)$$

where $x_{map(i)}$ denotes the road coordinates in the horizontal direction and $z_{map(i)}$ represents the road height coordinates. Equations (4a) and (4b) are used for calculating the height difference and the gradient between two points, respectively. If the height difference

deviates from a certain range, it implies that there is a topographic change, and the deviation of the gradient between two points from a certain range indicates a feature point of the parts showing large changes, i.e., edges. The edge points building the boundary between the asphalt and the lane can be easily obtained with this equation. In Equation (4a), d denotes the minimum height difference for a feature point, and σ in Equation (4b) represents the minimum angle between two points for a feature point.

4.2. Curvature Algorithm

When a point moves along a curve at a given velocity, the moving direction changes according to s , the distance covered (the length of the arc of the curve). Therefore, the rate of change is called the curvature of the curve. Figure 7 illustrates the degree of rotation required to follow the curve while moving along the curve from the start point to the end point.

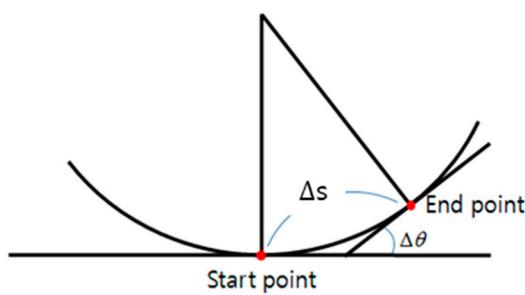


Figure 7. Curvature of the curve.

where Δs denotes the point-to-point infinitesimal distance and $\Delta\theta$ represents the infinitesimal angle formed by the tangent lines of two points.

$$k = \lim_{\Delta s \rightarrow \infty} \left| \frac{\Delta\theta}{\Delta s} \right| = \left| \frac{d\theta}{ds} \right| \quad (5)$$

where Δs denotes the infinitesimal distance covered from a start point on a curve to the end point and $\Delta\theta$ denotes the angle formed by two tangent lines at two points. Further, the multiplicative inverse of curvature is the curvature radius. The curvature radius is the radius of a circle having a certain curvature and is also referred to as the rotational radius. Curvature radius ρ is defined as Equation (6).

$$k > 0 \Rightarrow \rho = 1/k \quad (6)$$

Equation (5) shows that Δs , which denotes the total length of the arc moved along the curve, is proportional to the curvature radius. On the whole, curvature k is shown to be inversely related to Δs . If $k = 0$, the curvature radius is infinity, and thus, the curve becomes a straight line. In other words, a straight line can be thought to be a circle having an infinite radius.

If the robot moves rotationally, a rotational centroid is generated, and this point is the instantaneous center of rotation (ICR). ICR is a point at which the extension lines of the rotational axes of the wheels meet [34–37]. Since the axis of the two wheels of the mobile robot analogous to the vehicle wheels' moving tendency is positioned along the same line, ICR can be positioned anywhere on the wheel axis. Further, the relationship between the wheel velocity values and the robot-to-ICR distance can be expressed by a proportional equation such as Equation (7).

$$v_L : v_R = \rho - \frac{l}{2} : \rho + \frac{l}{2} \quad (7)$$

where v_L denotes the velocity of the left side wheels; v_R , that of the right side wheels; ρ , the curvature radius; and l , the lane between two wheels of the robot. The curvature radius can be expressed by Equation (8).

$$\rho = \frac{l}{2} \left(\frac{v_R + v_L}{v_R - v_L} \right) \quad (8)$$

In other words, the rotational radius is determined by the velocity of the robot's left and right wheels. According to the above equation, if the robot navigates along a straight line, $\rho = \infty$ ($v_R = v_L$), and if $v_R \neq v_L$, the robot implements a circular motion.

In Figure 8, if the mobile robot moves from point A to point B, let the time at point A be t_0 and its location (x_p, y_p, θ_p) , then the time at point B can be expressed as $t_1 = t_0 + \Delta t$ and its location as (x'_p, y'_p, θ'_p) . ICR's coordinates can be defined as in Equation (9).

$$ICR = [x_R - R \sin(\theta_R), y_R + R \cos(\theta_R)] \quad (9)$$

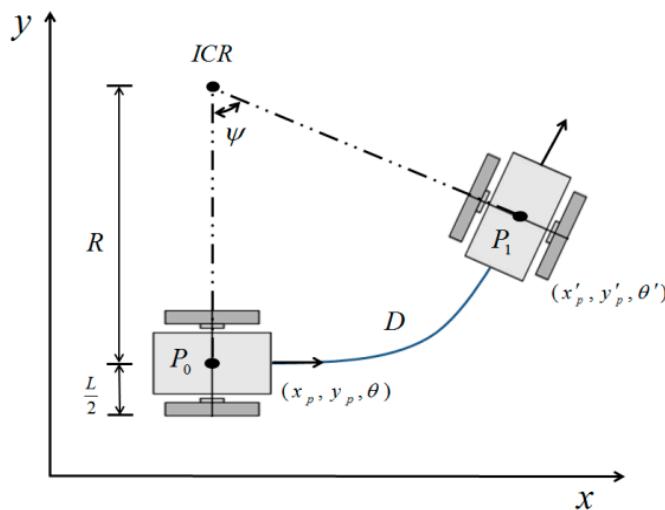


Figure 8. ICR of the mobile robot.

The robot's location (x'_R, y'_R, θ'_R) at the instant of time lapse of δt from t_0 can be expressed by Equation (10).

$$\begin{bmatrix} x'_p \\ y'_p \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\psi \bullet \Delta t) & -\sin(\psi \bullet \Delta t) & 0 \\ \sin(\psi \bullet \Delta t) & \cos(\psi \bullet \Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p - ICR_x \\ y_p - ICR_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICR_x \\ ICR_y \\ \psi \bullet \Delta t \end{bmatrix} \quad (10)$$

$$= \begin{bmatrix} R \sin(\psi \bullet \Delta t) + ICR_x \\ -R \cos(\psi \bullet \Delta t) + ICR_y \\ \theta + \psi \bullet \Delta t \end{bmatrix}$$

Equation (10) expresses the mobile robot's location as an equation governing the location of ICR and the angular velocity. The numerical expressions of the total moving distance, D , of the mobile robot from P_0 to P_1 and the rotation angle, ϕ , are expressed as follows:

$$D = \int_t^{t+\Delta t} V dt = \int_t^{t+\Delta t} \frac{v_l + v_r}{2} dt \quad (11)$$

$$\phi = \frac{D}{R} = \frac{\int_t^{t+\Delta t} (v_l + v_r) dt}{L(v_l + v_r)} \bullet (v_r - v_l) \quad (12)$$

4.3. Limitations of Mobile Robot

The velocities of the mobile robot's left and right wheels are denoted as v_L and v_R , and their accelerations are denoted as a_L and a_R , respectively, as in the following equations.

$$|v_L|, |v_R| \leq v_{\max} \quad (13)$$

$$|a_L|, |a_R| \leq a_{\max} \quad (14)$$

where v_{\max} refers to the maximum design velocity of the mobile robot, whereas a_{\max} refers to the maximum design acceleration. In this study, v_{\max} was set to 0.1 m/s and a_{\max} was set to 0.1 m/s².

4.4. Lane Tendency and Lane Prediction Using a Curvature Algorithm

In general, as explained above, curvature is used for determining the robot's path or time-dependent location in cases where the start and end points as well as the robot's navigating and angular velocities are clearly known. The curvature radius is calculated using the start and target points as well as the robot's direction angle, and the robot's navigating path is figured out using the curvature radius and the robot's velocity.

However, the method proposed in this paper was used for predicting the navigation path of the mobile robot by using a curvature algorithm, in cases where the lanes became recognizable while driving, by applying the feature points recognized up to the start point of the discontinuity and using the recognized tendency. Thus, this method has no preset target point. Equation (15) is used for figuring out the lane tendency up to the point of discontinuity in order to predict the future course of the lanes. It expresses the point-to-point distances at the $n - 1$ th point and the change rate of the yaw-axis angle.

$$(D_{ave}, \theta_{ave}) = \left(\frac{1}{n-1} \sum_{a=2}^n (\sqrt{x_a - x_0})^2 + (\sqrt{y_a - y_0})^2, \frac{1}{n-1} \sum_{a=2}^n \theta_a \right) \quad (15)$$

where D_{ave} denotes the average distance; θ_{ave} , the average change rate of the yaw-axis angle; x_0 , the $n - 1$ th x-axis coordinates; y_0 , the $n - 1$ th y-axis coordinates; and θ_a , the amount of angular change between the $n - 1$ th point and the a th point. The lane tendency thus obtained can be calculated to yield the curvature radius by using Equation (16).

$$\rho = \frac{D_{ave}}{2 \cos(\theta_{ave} + 180)} \quad (16)$$

The navigation path can be predicted using the calculated curvature radius and the point-to-point distance.

$$x_{est} = \rho \cos(\theta_{ave}) \quad (17a)$$

$$y_{est} = \rho \sin(\theta_{ave}) \quad (17b)$$

Equations (17a) and (17b) is used for calculating the x and y coordinates at the point where the lanes are not visible; the robot can move along these coordinates.

4.5. Path Monitoring through Update

A dashed line is divided into lined and unlined parts. In the unlined parts, the robot's navigation path is established by using the curvature-theory-based lane prediction. Wherever lines are detected, navigation follows the visible lines. However, in the case of a dashed line, lines are not found within the given interval. In this case, curvature is not generated using Equations (15)–(17), but a more accurate curvature can be generated using the curvature generated immediately before finding the line and the data of the dashed line covered up to the point as well as Equations (15)–(17). Figure 9 shows the flow of the entire system.

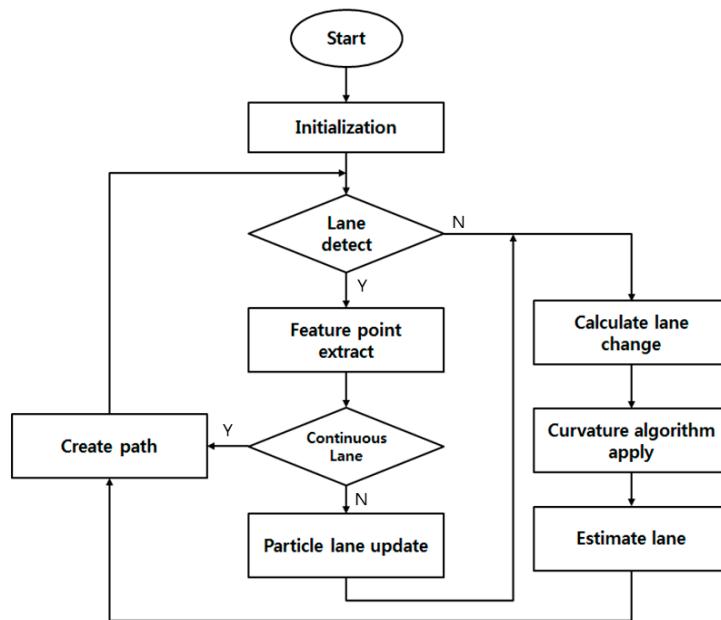


Figure 9. Total system flow.

After the initialization, the system scans the road using LRF. Lanes are then sought using the feature point extraction algorithm. When the feature points of the lane are extracted and the lane borders are found, a path is generated via the line coordinates. If no lines are found, the changing line pattern is figured out using the line coordinates available up to that point. Thus, a path is generated using the curvature algorithm as well as the lane change rate. When the line recognized is not a continuous line, but a dashed line, the curvature algorithm is updated with the information on the given line, and a more accurate prediction path can be generated.

5. Tracking Control Algorithm

This algorithm is a stable tracking control algorithm to progressively control the Cartesian trajectory from the start point to the target point of an off-trajectory robot using the mobile robot's present location and position and the location of the target point. The location of the robot is expressed using three degrees of freedom as in Equation (18) where x and y denote the robot's coordinates, which can be obtained via the encoder values of the motors on both sides, and θ denotes the robot's eye direction.

$$P_k = [x, y, \theta]^T \quad (18)$$

Further, all terms of Equation (14) are time functions. Therefore, they can be expressed as $x(t)$, $y(t)$, and $\theta(t)$, and $\dot{\theta}(t)$ can be obtained using Equation (15).

$$\theta(t) = \tan^{-1} \left(\frac{\dot{y}(t)}{\dot{x}(t)} \right) \quad (19)$$

The motions of the mobile robot are controlled with linear and angular velocities. The mobile robot's kinematics is defined by using the Jacobian matrix J .

$$\dot{P} = Jq \quad (20)$$

where q denotes $[v \ w]^T$ and J is $\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$. This kinematics algorithm can be applied commonly to all non-omnidirectional robots. This algorithm requires the locations

of the current and target points in order to control a mobile robot. The system calculates P_e , the difference between the two points, using Equation (20) [37].

$$P_e = T_e(P_r - P_c) \quad (21)$$

where T_e is $\begin{bmatrix} \cos \theta_c & \sin \theta_c & 0 \\ -\sin \theta_c & \cos \theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and θ_c denotes the present direction angle of the robot.

Finally, the linear and angular velocities required for the robot to move up to the target point can be calculated via Equation (18).

$$q = \begin{pmatrix} v_d \cos \theta_e - v_c \\ \omega_d - \omega_c \end{pmatrix} \quad (22a)$$

$$v_d = \pm \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (22b)$$

$$\omega_d = \frac{\ddot{y}_r(t)\dot{x}_r(t) - \ddot{x}_r(t)\dot{y}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (22c)$$

where v_d, ω_d denote the linear and angular velocities required to reach the target point, v_c, ω_c represent the present linear and angular velocities, and ω_c, y_r denote the coordinates of the target point.

Figure 10 shows the structure of the following controller. $p_r(t)$ represents the target position and $p_c(t)$ represents the current position of the mobile robot. Through the error matrix, which is the difference p_e between the current position and the target position, q composed of the required linear velocity and angular velocity is obtained. The third box T is converted to fit the hardware considering the structure of the hardware. The coordinates of the mobile robot reflecting the required speed are combined with the Jacobian J and changed to the current speed which is accumulated and feedback to the current position.

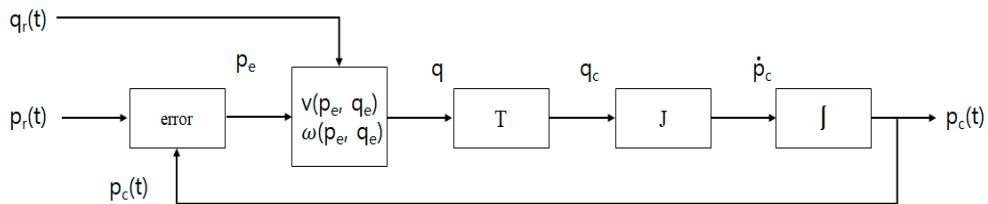


Figure 10. The architecture of the tracking controller.

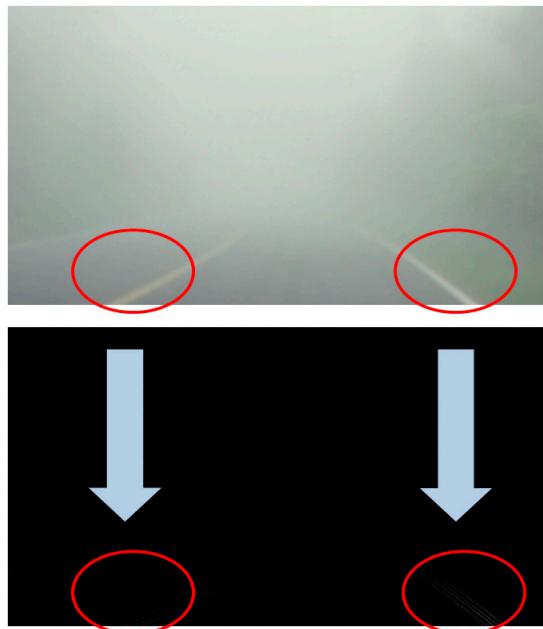
6. Experiment

6.1. Comparison Using Vision

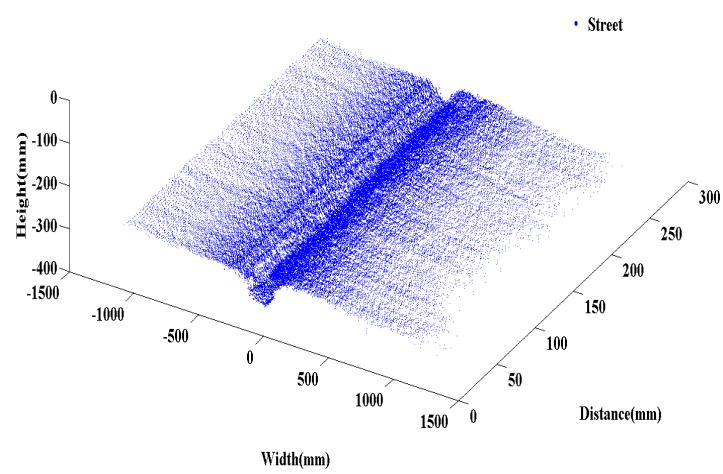
The first experiment involved driving under foggy conditions in a comparison test of a vision-based system and LRF to demonstrate that the vision-based system is influenced by fog whereas LRF is not. The first picture in Figure 11a shows the test environment on a foggy road. The picture beneath it shows the vision-based processing. Figure 11b shows an onsite LRF-scanned 3D road map.

In order to extract the lane-related information from the image obtained using the vision-based system, various filters or considerable computations are required. We extracted the lane information using SOBEL after changing the image into a gray-scaled one. As shown in the picture above, the front lane is partially visible as a blurred image, but the overall visibility for extracting the lane information is very limited. Moreover, under more severe foggy conditions, it would be a huge challenge to find lanes using vision. In contrast, LRF was not influenced by the fog at all, demonstrating the same effect as that

on a clear day. Figure 11b shows a 3D road map built using the LRF scan data around the center line of the lane. The robot was instructed to navigate at the velocity of 1 m/s.



(a) Test environment (above); vision-based system (below)



(b) LRF-scanned 3D road map

Figure 11. Lane detection using vision in foggy weather.

In Figure 12, the lane feature points extracted from the 3D road map shown in Figure 11b by using the feature point extraction algorithm are highlighted with yellow dots. The robot was instructed to navigate along the right-hand-side boundary line. The errors in the case of the real lane width and the case of the scanned lane width with respect to the right-hand-side boundary line were compared.

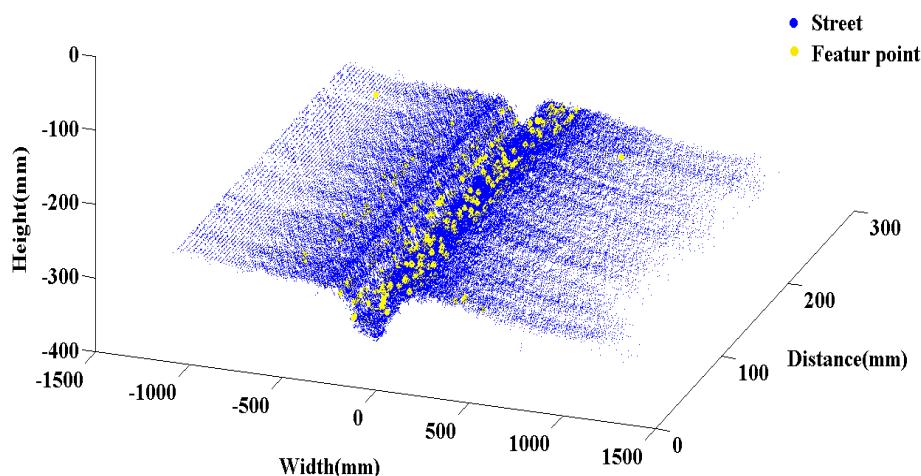


Figure 12. Feature point extraction.

In Table 5, Given the land width of 150 mm, the average feature point coordinates did not deviate from those of the real lane width. Further, the maximum error was below 20 mm, not largely deviating from the real lane width. Nor did the robot deviate considerably from the lane during navigation because of the data error.

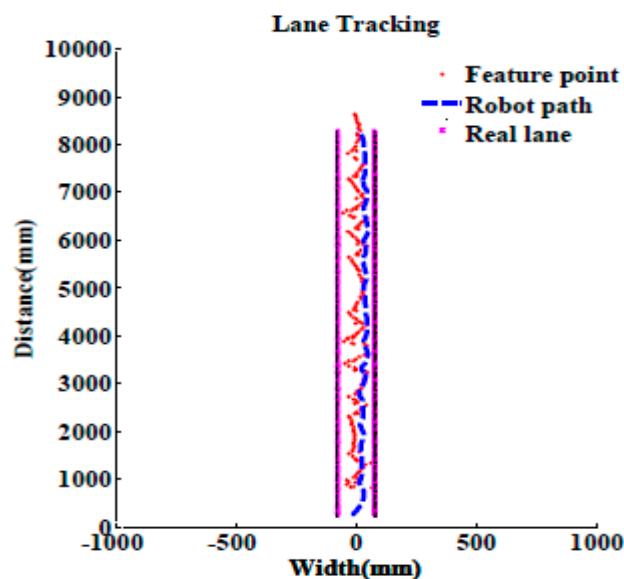
Table 5. Lane width error of feature point extraction.

	Max (mm)	Mean (mm)	Max Error (mm)
Real lane	-	150	-
LRF-scanned lane	169	104	19 mm (12.7%)

Furthermore, we conducted the same experiment on a dark road at night without street lamps and obtained the same results as those shown in Figure 11b. The results of this experiment verified that LRF could recognize lanes irrespective of the weather or light conditions by yielding the same results on foggy or dark roads as those on clear and bright roads.

6.2. Stable Tracking Algorithm Using Lane Tracking

The second experiment is an experiment that allows mobile robots to follow the lanes found on the three-dimensional map. In Figure 13, the red dots represent the densest points among the singularities of the lane and the part marked with the blue dotted line indicates the path the mobile robot moved. The purple straight line of both sides shows the real lane. The results of this experiment show that the detected lane is stably driven using a stable following algorithm.

**Figure 13.** Result of lane tracking test.

Furthermore, the singularity of the lane is zigzag, which calculates the singularity density of both lane edges, so it can appear irregularly in the lane according to the density of the detected singularity.

Figure 14 shows a graph of error. The blue solid line is the detection of the singularity of the actual lane, and the mobile robot must follow this line while driving at 1 m/s, but an error occurred as shown in the red dotted line. The maximum tracking error was 147.65 mm. However, when looking at the overall driving as shown in Figure 14, it can be seen that the robot's driving path was stably driven without departing from the lane.

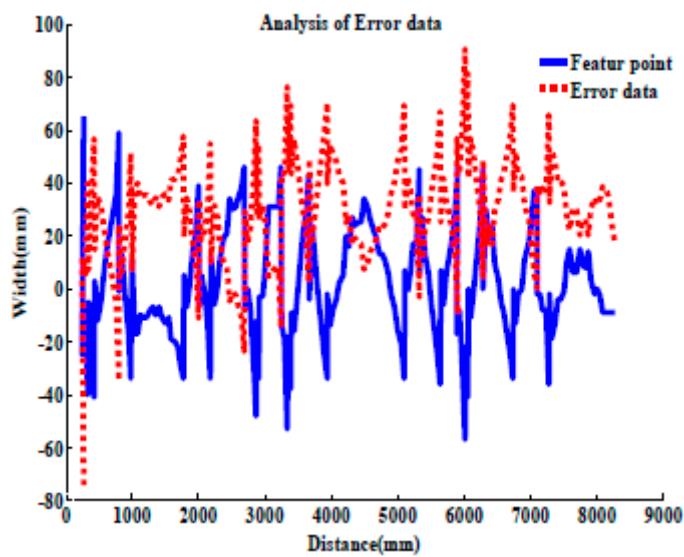


Figure 14. The analysis of error data.

6.3. Lane Prediction Using the Curvature Algorithm

We tested whether the driving path could be predicted by applying the curvature algorithm to a dashed-line curved path when the lines could not be scanned.

Figure 15 shows the 3D road map (right) of the test environment (left). Figure 16 shows the feature points extracted from this map.

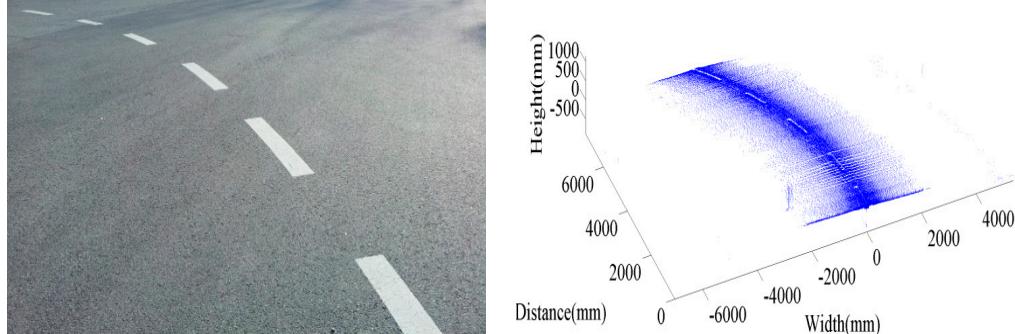


Figure 15. The 3D Road map of a discontinuous lane.

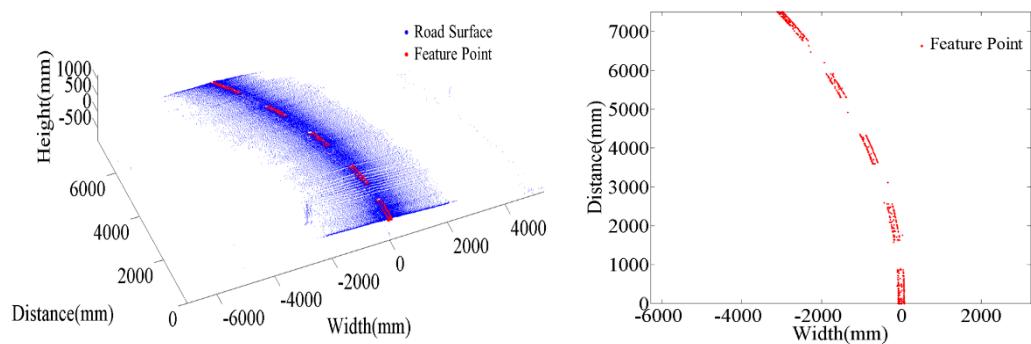


Figure 16. Feature points of a discontinuous lane.

If the curvature algorithm is applied using the feature points extracted as shown in Figure 16, the path can be predicted as shown in Figure 17.

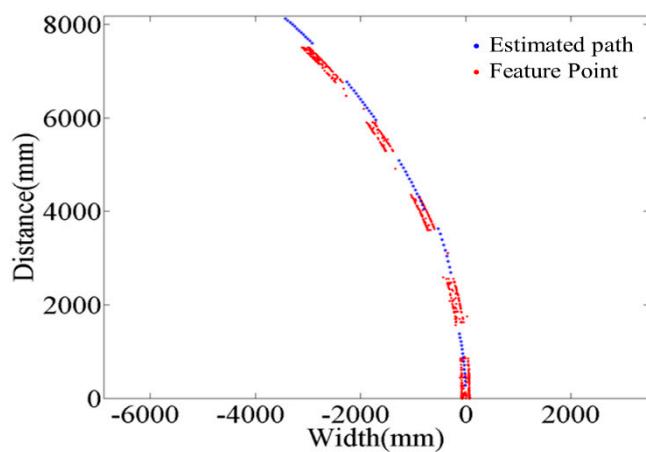


Figure 17. Application of the curvature algorithm.

As shown in Figure 18, the driving path can be predicted with blue dots by using the curvature of the road sections that do not have any marked lines. The predicted dots do not deviate considerably from the lane and provide the robot with a path to follow that is almost analogous to the lane. Figure 15 shows that the robot navigates well without deviating from the dashed-line lane.

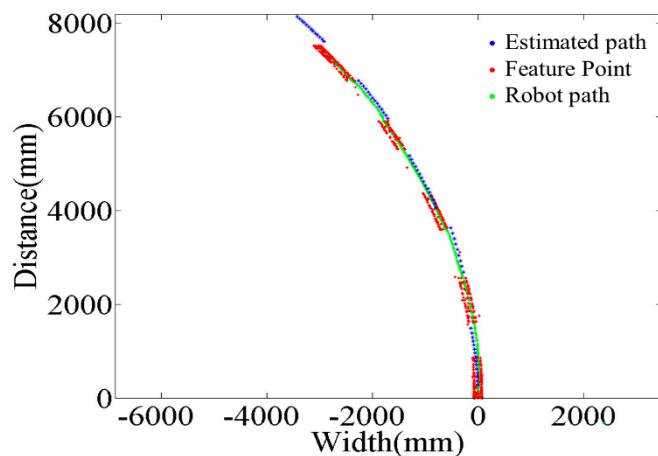


Figure 18. Predicted path for driving.

As shown in Figure 19a, we tested whether lane prediction can be implemented by using curvature in cases where the line markings are partly unrecognizable. Figure 19b shows the prediction test result for the test environment shown in Figure 19a when the algorithm was not used. It shows that the prediction was not correct after the first dashed-line lane and the robot failed to find the dashed-line lane again. In contrast, Figure 19c clearly demonstrates that by applying the curvature algorithm, lane prediction is possible even if the marked lines are partially unrecognizable.

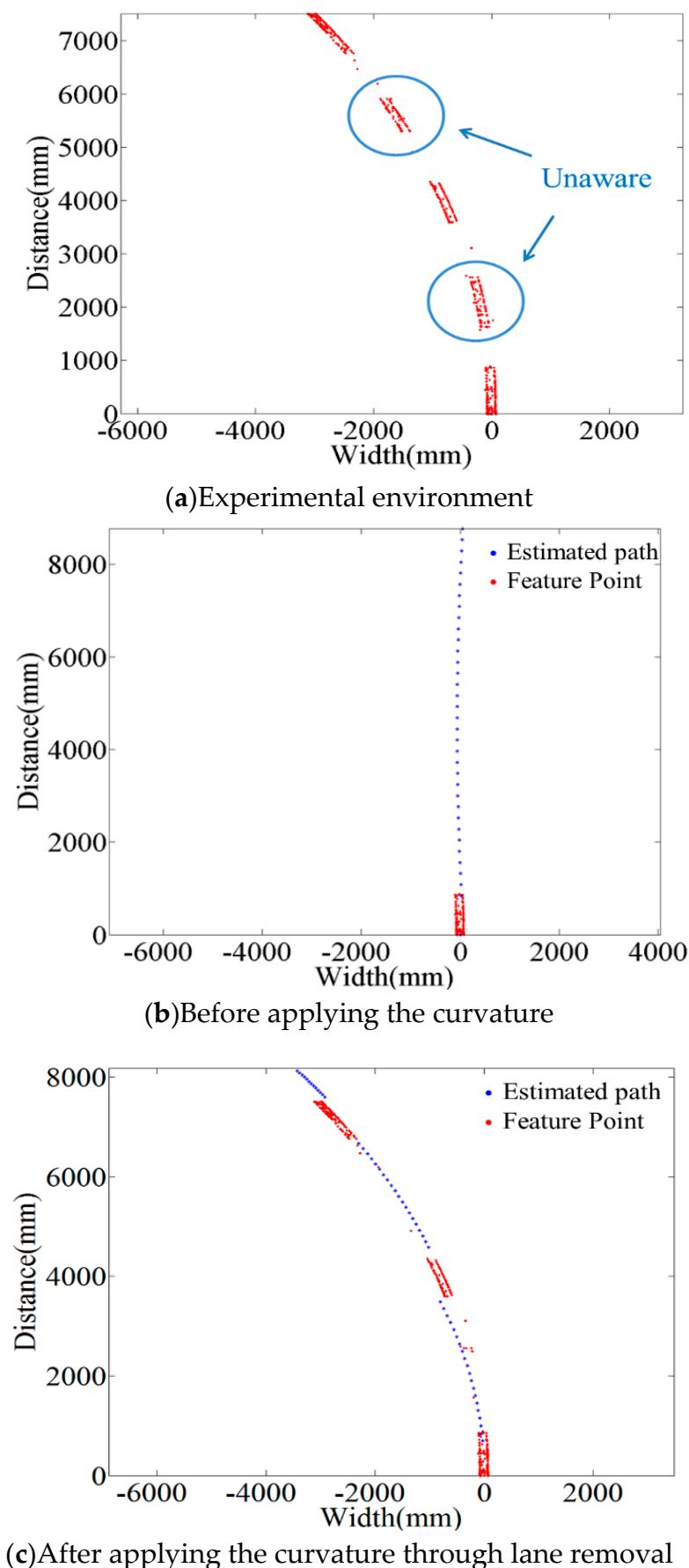


Figure 19. Lane prediction when some lanes are not recognizable.

7. Conclusions

This paper proposed a real time lane detection algorithm using LRF (Laser Range Finder) for autonomous navigation of a mobile robot. By the vision-based system, recognition of the environment for three dimensional space becomes excellent only in good conditions for capturing images. However, there are so many unexpected barriers, such as bad illumination, occlusions, and vibrations that the vision cannot be used for satisfying the fundamental requirement, and conventional lane detection has mainly been carried out by using vision-based methods, but such methods have a serious drawback of showing substantially diminished performance in driving environments where reliable vision-based information is not obtained, such as under conditions of dense fog. Therefore, in this study, we built a laser-scanned 3D road map and discerned and recognized lanes by using a feature point extraction algorithm using LRF calibration and amplification errors depending on the materials and colors of the asphalt and the lanes. The test results confirmed that the use of the proposed method could ensure safe driving under unfavorable road conditions such as fog, which could contribute to the R&D on autonomous driving technologies. As future research, we plan to have a variety of tests in different environmental conditions and we have to apply a new method (RANSAC, MLESAC and so on).

Author Contributions: J.-H.H. collected and merged the data into a central database before analysis, H.-W.K. designed the figures. All other authors were involved and wrote the draft of the manuscript and in revising the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not available.

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

References

1. Sparbert, J.; Dietmayer, K.; Streller, D. Lane detection and street type classification using laser range images. In Proceedings of the 2001 IEEE Intelligent Transportation Systems, Proceedings (Cat. No. 01TH8585), Oakland, CA, USA, 25–29 August 2001.
2. Kim, Y.J.; Lee, K.B. *Lateral Control of Vision-Based Autonomous Vehicle Using Neural Network*; Korea Society for Precision Engineering: Seoul, Korea, 2000.
3. Kim, S.-H.; Kim, M.-J.; Kang, S.-C.; Hong, S.-K.; Roh, C.-W. Development of Patrol Robot using DGPS and Curb Detection. *J. Control. Autom. Syst. Eng.* **2007**, *13*, 140–146. [[CrossRef](#)]
4. Pan, Z.; Chen, H.; Li, S.; Liu, Y. Cluster Map Building and Relocalization in Urban Environments for Unmanned Vehicles. *Sensors* **2019**, *19*, 4252. [[CrossRef](#)]
5. Jung, J.-W.; Park, J.-S.; Kang, T.-W.; Kang, J.-G.; Kang, H.-W. Mobile Robot Path Planning Using a Laser Range Finder for Environments with Transparent Obstacles. *Appl. Sci.* **2020**, *10*, 2799. [[CrossRef](#)]
6. Jiang, G.; Yin, L.; Liu, G.; Xi, W.; Ou, Y. FFT-Based Scan-Matching for SLAM Applications with Low-Cost Laser Range Finders. *Appl. Sci.* **2018**, *9*, 41. [[CrossRef](#)]
7. Kluge, K.; Lakshmanan, S. A deformable-template approach to lane detection. In Proceedings of the Intelligent Vehicles '95. Symposium, Detroit, MI, USA, 25–26 September 1995.
8. Serge, B.; Michel, B. Lane Detection for Automotive Sensor. *JCASSP* **1995**, *5*, 2955–2958.
9. Dickmanns, E.D.; Graefe, V. Dynamic Monocular Machine Vision and Application of Dynamic Monocular Vision. *Int. J. Mach. Vis. Appl.* **1988**, *1*, 223–240. [[CrossRef](#)]
10. Kreucher, C.; Lakchmanan, S. LANA: A lane extraction algorithm that uses frequency domain features. *IEEE Trans. Robot. Autom.* **1999**, *15*, 343–350. [[CrossRef](#)]
11. Risack, R.; Klausmann, P.; Kruger, W.; Enkelmann, W. Robust Lane Recognition Embedded in a Real-Time Driver Assistance System. In *Proceedings of the Intelligent Vehicles Symposium, Stuttgart, Germany, 28–30 October 1998*; IEEE Press: Piscataway, NJ, USA, 1998; Volume 1, pp. 35–40.
12. Wang, Y.; Shen, D.; Teoh, E.K. Lane Detection using Catmull-Rom Spline. In *Proceedings of the Intelligent Vehicles Symposium, Stuttgart, Germany, 28–30 October 1998*; IEEE Press: Piscataway, NJ, USA, 1998; Volume 1, pp. 51–57.
13. Suh, S.B.; Kang, Y.; Roh, C.-W.; Kang, S.-C. Experiments of Urban Autonomous Navigation using Lane Tracking Control with Monocular Vision. *J. Inst. Control. Robot. Syst.* **2009**, *15*, 480–487.
14. Eun, S. Enhancing the Night Time Vehicle Detection for Intelligent Headlight Control using Lane Detection. In Proceedings of the KSSE 2011 Annual Conference, Topeka, KS, USA, 8–9 December 2011.
15. Montemerlo, M.; Beeker, J.; Bhat, S.; Dahlkamp, H. The Stanford Entry in the Urban Challenge. *J. Field Robot.* **2008**, *7*, 468–492.

16. Chun, C.-M.; Suh, S.-B.; Lee, S.-H.; Roh, C.-W.; Kang, S.-C.; Kang, Y.-S. Autonomous Navigation of KUVE (KIST Unmanned Vehicle Electric). *J. Inst. Control. Robot. Syst.* **2010**, *16*, 617–624. [[CrossRef](#)]
17. Jung, B.-J.; Park, J.-H.; Kim, T.-Y.; Kim, D.-Y.; Moon, H.-P. Lane Marking Detection of Mobile Robot with Single Laser Rangefinder. *J. Inst. Control. Robot. Syst.* **2011**, *17*, 521–525. [[CrossRef](#)]
18. Hiroi, Y.; Ito, A. A Pedestrian Avoidance Method Considering Personal Space for a Guide Robot. *Robotics* **2019**, *8*, 97. [[CrossRef](#)]
19. Ali, M.A.; Mailah, M.; Jabbar, W.A.; Moiduddin, K.; Ameen, W.; Alkhalefah, H. Autonomous Road Roundabout Detection and Navigation System for Smart Vehicles and Cities Using Laser Simulator-Fuzzy Logic Algorithms and Sensor Fusion. *Sensors* **2020**, *20*, 3694. [[CrossRef](#)] [[PubMed](#)]
20. Li, L.; Luo, W.; Wang, K.C.P. Lane Marking Detection and Reconstruction with Line-Scan Imaging Data. *Sensors* **2018**, *18*, 1635. [[CrossRef](#)] [[PubMed](#)]
21. Assuncao, A.N.; Aquino, A.L.L.; Santos, R.C.C.D.M.; Guimaraes, R.L.M.; Oliveira, R.A.R. Vehicle Driver Monitoring through the Statistical Process Control. *Sensors* **2019**, *19*, 3059. [[CrossRef](#)]
22. Cao, J.; Song, C.; Song, S.; Xiao, F.; Peng, S. Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments. *Sensors* **2019**, *19*, 3166. [[CrossRef](#)] [[PubMed](#)]
23. Hoang, V.-D.; Jo, K.-H. A Simplified Solution to Motion Estimation Using an Omnidirectional Camera and a 2-D LRF Sensor. *IEEE Trans. Ind. Inform.* **2016**, *12*, 1064–1073. [[CrossRef](#)]
24. DU, X.; Tan, K.K. Comprehensive and Practical Vision System for Self-Driving Vehicle Lane-Level Localization. *IEEE Trans. Image Process.* **2016**, *25*, 2075–2088. [[CrossRef](#)] [[PubMed](#)]
25. Wang, H.; Wang, Y.; Zhao, X.; Wang, G.; Huang, H.; Zhang, J. Lang Detection of Curving Road for Structural High-way with Straight-curve Model on Vision. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5321–5330. [[CrossRef](#)]
26. Dai, K.; Wang, Y.; Ji, Q.; Du, H.; Yin, C. Multiple Vehicle Tracking Based on Labeled Multiple Bernoulli Filter Using Pre-Clustered Laser Range Finder Data. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10382–10393. [[CrossRef](#)]
27. SICK. Available online: <http://www.sick.com> (accessed on 25 October 2019).
28. Hokuyo Automatic Co., Ltd. Available online: <http://www.hokuyo-aut.co.jp> (accessed on 25 October 2019).
29. Kawata, H.; Miyachi, K.; Hara, Y.; Ohya, A.; Yuta, S. A method for estimation of lightness of objects with intensity data from SOKUIKI sensor. In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Seoul, Korea, 20–22 August 2008; pp. 661–664. [[CrossRef](#)]
30. Kawata, H.; Ohya, A.; Yuta, S.; Santosh, W.; Mori, T. Development of ultra-small lightweight optical range sensor system. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 1078–1083. [[CrossRef](#)]
31. Zhu, J.; Ahmad, N.; Nakamoto, H.; Matsuhira, N. Feature Map Building Based on Shape Similarity for Home Robot ApriAtenda. In Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics; Institute of Electrical and Electronics Engineers (IEEE), Kunming, China, 17–20 December 2006; pp. 1030–1035.
32. Jia, S.; Yang, H.; Li, X.; Fu, W. LRF-based data processing algorithm for map building of mobile robot. In Proceedings of the 2010 IEEE International Conference on Information and Automation, Harbin, China, 20–23 June 2010; pp. 1924–1929.
33. Grossman, S.I.; Derrick, W.R. *Advanced Engineering Mathematics*; Harper Collins Publishers: New York, NY, USA, 2000.
34. Dudek, G.; Jenkin, M. *Computational Principles of Mobile Robotics*; Cambridge University Press (CUP): Cambridge, UK, 2009.
35. Kim, M.S.H.; Han, J.; Lee, J. Optimal Trajectory Control for Capturing a Mobile Sound Source by a Mobile Robot. *Int. J. Hum. Robot.* **2017**, *14*, 1750025. [[CrossRef](#)]
36. Han, J.-H. Tracking Control of Moving Sound Source Using Fuzzy-Gain Scheduling of PD Control. *Electronics* **2019**, *9*, 14. [[CrossRef](#)]
37. Kanayama, Y.; Kimura, Y.; Miyazaki, F.; Noguchi, T. A stable tracking control method for an autonomous mobile robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 1, pp. 384–389. [[CrossRef](#)]