This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Hybrid tracker based optimal path tracking system of autonomous driving for complex road environments

EUNBIN SEO[1], SEUNGGI LEE[1], GWANJUN SHIN[1], HOYEONG YEO[1], YONGSEOB LIM[2], AND GYEUNGHO CHOI[3]

[1]Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, South Korea (e-mail: {seb5428, seunggi_lee, shinkansan, hoyeong23}@dgist.ac.kr)
[2]Department of Robotics Engineering, Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, South Korea
[3]Department of Interdisciplinary Engineering, Daegu Gyeognbuk Institute of Science and Technology, Daegu 42988, South Korea

Corresponding authors: Yongseob Lim (yslim73@dgist.ac.kr) and Gyeungho Choi (ghchoi@dgist.ac.kr)

This research was conducted by the DGIST institution-specific project supported by the Ministry of Science and ICT (21-BRP-09 & 21-BRP-08).

ABSTRACT Path tracking system plays a key technology in autonomous driving. The system should be driven accurately along the lane and be careful not to cause any inconvenience to passengers. To address such tasks, this research proposes hybrid tracker based optimal path tracking system. By applying a deep learning based lane detection algorithm and a designated fast lane fitting algorithm, this research developed a lane processing algorithm that shows a match rate with actual lanes with minimal computational cost. In addition, three modified path tracking algorithms were designed using the GPS based path or the vision based path. In the driving system, a match rate for the correct ideal path does not necessarily represent driving stability. This research proposes hybrid tracker based optimal path tracking system by applying the concept of an observer that selects the optimal tracker appropriately in complex road environments. The driving stability has been studied in complex road environments such as straight road with multiple 3-way junctions, roundabouts, intersections, and tunnels. Consequently, the proposed system experimentally showed the high performance with consistent driving comfort by maintaining the vehicle within the lanes accurately even in the presence of high complexity of road conditions. Code will be available in https://github.com/DGIST-ARTIV.

INDEX TERMS Intelligent vehicles, Vehicle driving, Autonomous vehicles, Path tracking, Lane detection, Driving stability.

I. INTRODUCTION

In recent years, the requirements for the driving system of the autonomous vehicle are increasing in difficulty with respect to levels of driving automation proposed by SAE [1]. In level 4-5 of vehicle autonomy, the vehicle should perform the main driving and request humans to transfer control only in special circumstances or in areas where autonomous driving is not possible. For this reason, combining the two fields of perception and control algorithm is emerging as a very important area in autonomous driving. Hence, There are several prominent researches about path tracking on autonomous driving [2], [3]. Recently, end-to-end deep learning tracking algorithms [4]–[6] or methods using reinforcement learning [7] have been studied, but these methods are difficult to response flexible changes in the surrounding environment.

The deep learning based lane detection and the GPS based driving path tracking were also applied to deal with complex road environments [8]. The control method used geometric model-based path tracker, Pure pursuit and Stanley controller, which are widely used in the control field. This research proposes a path tracking system that is able to pass through not only highways and motorway roads, but also urban and complex road environments. In high-speed and low-speed, tunnel, and steep curve, the system quickly changed the appropriate tracking method and reflected it in driving stability for different speeds. In lane detection, the algorithm designed to process the deep learning method has been applied the three lane fitting methods in parallel to provide the most optimized path.

In this research, the modified path tracking algorithm is

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

designed to simultaneously process vision data to GPS data, in other words, local and global coordinates. Vision and GPS based driving guidance lines cause uncomfortable driving textures when driving along the lines. Accurately following the lane is also an act that does not take into account driving stability or driver acceptance. To solve this problem, this research suggested an optimal path tracking algorithm to ensure high driving stability and comfortableness in the presence of high complexity of road conditions.

This optimal path tracking algorithm appropriately selects the most stable tracker out of modified trackers, which is called *Hybrid* tracker based optimal path tracking system in this research. This system receives sensor information at the same time as each tracker calculation, estimates the driving environment, and transfers the control authority to the most suitable tracker. With parallel system to minimize the stability degradation. This method guarantees consistent driving stability even in the presence of various driving environments with different characteristics. Furthermore, it has the advantage that additional correction of the computational speed is not required on the hybrid system even if each tracker is modified. Consequently, this research notes that there are several advantages of using the *Hybrid* tracker based optimal tracking system: (a) it is capable of driving in a complex road environment with high performance of driving stability and accuracy; (b) additional correction of the computational speed is not required even if each tracker is modified; (c) the proposed algorithm enhances high usage of vision based lane following which can be widely used in real-world environments; (d) Finally, our selection system induces the improvements of driving stability and tracking performance despite of its simple implementation. To demonstrate the generality of proposed system, we installed the system in an autonomous vehicle and were confirmed its performance at the DGIST campus and high-speed driving proving ground. Consequently, the experimental results represents that our proposed system significantly improves accuracy and stability on all seven cases at a low computational cost.

This research is structured as follows: Section II describes related works. Section III-B shows lane processing algorithm using result from deep learning. Section III-C introduces the modified path trackers and respective features. Section III-D describes the coordinate system transformation and interpolation for the path from the perception part. Section III-E demonstrates path tracker selection. Section IV shows experimental results and discussions of path trackers and proposed system. Finally, Section V provides the conclusion.

II. RELATED WORKS

A. PATH TRACKING ALGORITHM

Generating a path that the vehicle should take to its destination and following the generated path are the most essential parts of an autonomous vehicle. There are three types of path tracking algorithms, which are divided into geometric model-based, kinematic model-based, and dynamic model-based algorithm [9]. Among these three path tracking algorithms, the simplest algorithm type is the geometric model-based path tracking algorithm such as Pure pursuit, Stanley controller, and Vector pursuit.

The algorithms used in this research are Pure pursuit and Stanley controller. Pure pursuit and Stanley controller use the same 2-dimension bicycle model regardless of vehicle type, so that the calculation is simple and easy to apply compared to other path tracking algorithms [10], [11]. Unfortunately, it was reported that these two algorithms work well only on low-speed and general roads [12]. Conversely, when the curvature of the road turns to large or the speed of the vehicle becomes high, it is unable to follow the generated path. Therefore if both kinematic model and dynamic model are used, tracking performance can be improved in more diverse environments. However, the above simple algorithms (*i.e.*, Pure pursuit and Stanley controller) were appropriately modified in order to achieve maximum efficiency with limited computer performance.

Pure pursuit's tracking performance depends on the look-ahead distance, which leads importantly to set an appropriate look-ahead distance. Otherwise, unlike other path tracking algorithms, Stanley controller does not use a look-ahead distance, so parameters have not to be tuned [13]. Comparing multiple path tracking algorithms [9], Pure pursuit shows robustness to low-speeds and large errors, but it does not work well at high-speeds [14], [15]. However, Stanley controller has a faster rate of convergence to the ideal path than the Pure pursuit, but there are cases where the steering angle sometimes diverges on a road with a large curvature. Thus, this study focuses on modifying the existing algorithms to minimize the disadvantages mentioned above and to validate the modified tracking algorithm in various situations.

B. LANE DETECTION NETWORKS

Studies about end-to-end network performance for lane detection are as follows. PointLaneNet [16] redefined the lane structure and lane distance to facilitate the training of the network. PINet [17] combined point estimation and point instance segmentation, but there were limitations in the presence of the local occlusions or unclear lanes. The lightweight model ENet-SAD [18] has been proposed by applying self attention distillation (SAD) methods to the existing ENet [19]. SCNN [20] and ENet-SAD obtain diverse and rich contextual information to solve these limitations. In the autonomous driving system, various methods are attempted to reduce the delay from sensor recognition to vehicle maneuvering as much as possible. Recently, deep learning is essentially considered in the perception field, but it has also disadvantages that it is not able to secure both processing speed (*i.e.*, inference time) and performance (*i.e.*, accuracy) due to the limitation of GPU hardware. However, processing performance varies depending on the size of the network parameter or the computational complexity of the layer. The recent deep learning methodologies have proposed various methods to solve these obstacles. ENet-SAD improved the processing speed by applying self-attention approach. Its

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

processing speed secured 10 times faster comparing with SCNN. In addition, a method of actively using conventional computer graphics technology has been proposed to improve the performance of deep learning algorithms. Thus, this research proposed fast and accurate lane processing algorithm using the results of ENet-SAD. As a result, low-latency and highly reliable lane detection system has been designed and validated in the presence of diverse road conditions.

C. ENSEMBLE METHOD

Recognizing lanes in raw images and extracting information about lanes in segmentation images are necessary processes. In machine learning, ensemble method combines predictions from multiple models rather than a single model, so that the final result achieves better performance. There are various methods such as voting and bootstrap aggregating (Bagging) in the ensemble method. The voting method compares the results predicted by various algorithms for the same dataset and chooses the final result. There are also two types of voting: hard and soft voting. Hard voting simply chooses the results that received the most votes among the predictions from different models. Soft voting selects the highest result by summing the probability of the predicted results from different models. Moreover, Bagging is a method of voting to determine the final results predicted by one algorithm on various datasets sampled differently by allowing overlap. In Teow's handwriting recognition research, the best results came out by adopting soft voting [21]. Soft voting increased the accuracy and stability of aggregation as a result of prediction in signal segments [22]. Thus, this research adopted soft voting that can be parallel processing guaranteed fast computational speed in lane fitting algorithm.

III. APPROACH

A. SENSOR LAYOUT

The locations of each sensor are shown in Fig.1. Three sensors were used: camera, GPS, and IMU. Logitech's Brio 4K Pro Webcam for camera, Synerex's RTK GNSS MRP-2000 for GPS, and WitMotion's HWT901B for the IMU were utilized. The camera, GPS and IMU were installed at a distance of 155cm, 93cm, 80cm from the ground and 213cm, 82.5cm, 378cm from the front part, respectively shown in Fig. 1. The entire autonomous driving system of the test vehicle is equipped with one main computing unit and one discrete deep learning inference unit. Vehicle control and decision were operated by the embedded computer of Nuvo-8108GC, and a computer based on Titan Xp GPU for deep learning based lane detection algorithm was used. Detailed specifications for each sensor are listed in TABLE 1.

B. FAST OPTIMAL LANE PROCESSING ALGORITHM

1) Data Preparation

A numerous and high-quality dataset is required for deep learning network to avoid overfitting. First of all, ENet-SAD with the published lane dataset CULane [20] and TuSimple [23] have been trained in this research. ENet-SAD inferred the better results when trained on CULane containing urban, rural and highway road types. CULane contains a total of 133,235 images, including training set of 88,880, validation set of 9,675 and test set of 2,782. This study developed an annotation tool that had the same file structure as CULane, and gathered approximately 45,000 images for road types in Korea.

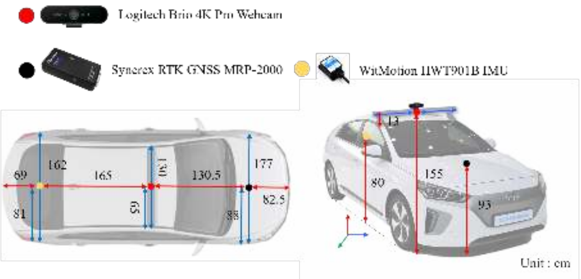


FIGURE 1. Sensor installation on the test vehicle.

TABLE 1. Specifications of sensors.

Sensor	Specification
Camera (Logitech Brio 4K Pro Webcam)	FoV 90 degree, up to 60 frame
Real Time Kinematic GPS (Synerex MRP-2000)	RTK Accuracy : 0.010m +1ppm CEP Fix time: 10 ∼ 60 sec
IMU (WitMotion HWT901B)	Angle Accuracy (after calibrated): X, Y-axis: 0.05°(Static), X, Y-axis: 0.1°(Dynamic)
Computer (Nuvo-8108GC)	CPU: Intel Xeon E 2176G, GPU: NVIDIA RTX 2080 Ti, 32GB RAM
Computer (Deep learning Inference)	CPU: Intel I7-9700F, GPU: NVIDIA TitanXp, 32GB RAM

The lane has linearity and the constant width of the lane. Since ENet-SAD learned the characteristics of the lane, the lane can be recognized even if there are objects or shadow covering the lane. The augmented data are also able to be obtained in a range that keeps the properties of lane. A total of 11 types of augmentation were applied without spatial information damage, including weather changes, brightness changes, shadow occlusion and resolution changes. As a result, total dataset consisted of 600,000 images with CULane and the Korean urban dataset. More dataset was obtained through augmentation, leading to the improved lane detection performance.

2) Post-process for Segmentation Image

The post-processing for lane segmentation images inferred from the ENet-SAD. The post-processing consists of inverse perspective mapping(IPM) and erosion filter. Due to the char-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

acteristic of camera, parallel lanes meet at the vanishing point and straight lines with the same thickness tend to become thinner as approaching the vanishing point. Images taken by camera have the distortion from the real circumstances described above and this distortion narrows the range of steering, which is able to impair steering sensitivity. To solve this phenomenon, IPM was used and it was possible to preserve the thickness and parallelism of straight lanes by converting to top view images. During converting the curve lanes, there is an area with thicker lanes which adversely affects the computational speed of the lane fitting process. The segmentation image with thin lanes could be obtained using the erosion filter. Therefore, this process leads to faster computational speed in lane fitting because it reduces the number of pixels that need to be processed. Lane fitting will be detailed in section III-B3. The Fig. 2 shows enlarged images of raw output from the ENet-SAD multiplied by a constant. The lanes are separated by different colors. Comparing the leftmost lane of Fig. 2(a) and Fig. 2(b), the thickest part is 66 pixels and 49 pixels, respectively. By simplifying data, the computational cost could be reduced.

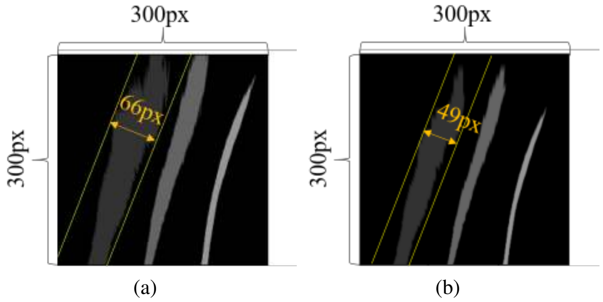


FIGURE 2. Before and after using erosion filter (a) segmentation image without erosion filter, (b) segmentation image with erosion filter.

3) Lane Fitting

The lane fitting finds the regression function that best represents the pixel coordinates detected as lane in the segmentation image. Specifically, the methods used in lane fitting are linear, quadratic and cubic least squares fitting. Equation (1) expresses a least squares polynomial fitting function. The coefficient values (*i.e.*, $c_i, i = 1, 2, \ldots, m$) for minimizing the squared error was derived from (1).

$$\begin{pmatrix} p_0(x_1) & p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ p_0(x_2) & p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_0(x_n) & p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{1}$$

where m means degree, $c_i(i = 1, 2, \ldots, m)$ is coefficients, $p_i(x)(i = 1, 2, \ldots, m)$ is x to the i power respectively.

The analyzed lanes consist of four lanes, which are defined as left-left lane, left lane, right lane, and right-right lane. Coefficient values were found by using linear, quadratic, and cubic least squares fitting of all four lanes. Some pixel coordinates were sampled by applying the coefficients obtained from the three fitting functions. Using the obtained variance as metric, the mentioned four lanes were expressed by selecting the fitting function with the smallest variance. Based on the fitting function that best represents lanes, 31 points are obtained for each lane, and the driving guidance line is provided in the form of a point group using the fitting function corresponding to the left lane and right lane. Although one of the two main lanes is not detected, driving guidance line (hereinafter referred to as a path) can be provided by adding an offset because a lane has certain width. The procedure for vision recognition parts configured the seamless system, allowing the tracker to process the path directly. The process of overall lane fitting algorithm can be seen in Fig. 3. Algorithm 1 shows a pseudo code that contains the overall fast optimal lane processing described in the section III-B.

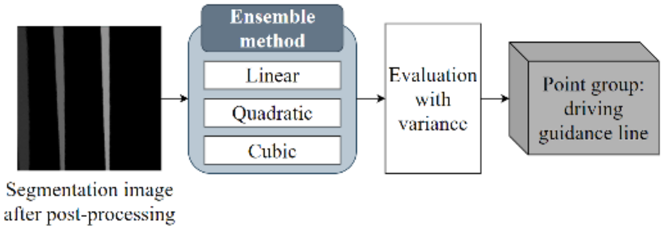

Ensemble method
Linear
Quadratic
Cubic
Evaluation with variance
Point group: driving guidance line
Segmentation image after post-processing

FIGURE 3. Procedures of the lane fitting algorithm.

C. PATH TRACKING ALGORITHM

1) Pure Pursuit

Pure pursuit has been recognized one of the path tracking algorithms that follow the target point. In this algorithm, a vehicle is treated as a bicycle model, and the geometric explanation of the Pure pursuit is shown in Fig. 4. The basic calculation method of Pure pursuit is as follows.

When drawing a circle with the radius of the look-ahead distance (ld) based on the rear wheel of the vehicle, the point that overlaps the given path is taken as the target point. The steering angle (δ) of the vehicle is calculated using α, which indicates the difference between the direction toward the rear wheel of the vehicle and the direction toward the look-ahead distance. At this time, the steering angle can be expressed as (2).

$$\delta = \tan^{-1}\left(\frac{2L \cdot sin(\alpha)}{ld}\right) \tag{2}$$

where, L represents wheel base of the vehicle and ld means the look-ahead distance respectively.

There are static and variable look-ahead distances. With a static look-ahead distance, the Pure pursuit achieves poor results because the target point is always fixed in the same

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

Algorithm 1 Overall fast optimal lane processing algorithm

```
Input: $I$: Segmentation image from deep learning network
Output: $O$: Driving guidance line consisting of points
$I_p$: Top view image after post-processing for $I$
$w$: Width of $I$
$h$: Height of $I$
$L_{coord}$: Two-dimensional lists of coordinates for lane points
$L_{func}$: Two-dimensional lists of function for lane
$I_p \leftarrow$ Run IPM
$I_p \leftarrow$ Apply the erosion filter
for $iteration = 1, 2, \ldots, w$ do
    for $iteration = 1, 2, \ldots, h$ do
        Get the pixel coordinates segmented by lane
        from $I_p$ and store them in the $L_{coord}$
    end for
end for
for $iteration(i) = 1, 2, 3, 4$ do
    Run lane fitting algorithm on the $L_{coord}[i]$
    Store the best function expression in the $L_{func}[i]$
end for
if left lane and right lane are exist then
    $O \leftarrow$ Calculate the center points with
    $L_{func}[1]$ and $L_{func}[2]$
else
    if only left is exist then
        $O \leftarrow$ Calculate the left lane
        points with $L_{func}[1]$ and add offset
    else
        if only right is exist then
            $O \leftarrow$ Calculate the right lane
            points with $L_{func}[2]$ and add offset
        else
            $O \leftarrow$ Lane does not exist
        end if
    end if
end if
```

radius regardless of the vehicle speed. Therefore, a variable look-ahead distance was used in order to achieve better results and expressed as a function of velocity in this study. The variable look-ahead distance was expressed as (3).

$$ld = 190 + 100 \cdot [\frac{1}{exp\{-(\frac{v-20}{15})\}} - 0.5] \tag{3}$$

Moreover, a PID controller based filter was added to prevent sudden bouncing or shaking of the steering angle. Anti-windup logic was added to prevent the divergence of error due to integral calculation. The compensator generating a smooth profile of the steering angle was also implemented as described in Algorithm 2.

In this study, the modified Pure pursuit was developed to follow the vision and GPS based path provided by the perception part. There are two types of pure pursuit depending on sensor used. Pure pursuit using a vision based path is indicated as Pure pursuit (vision), and Pure pursuit using a GPS based path is indicated as Pure pursuit (GPS). Both Pure pursuit (vision) and Pure pursuit (GPS) become unable to drive on very large curvature roads, but stability of steering angle is highly secured on straight or slight curve. Pure pursuit (vision) is more responsive than Pure pursuit (GPS) due to fast computational speed of vision. However, Pure pursuit (vision) is unable to compute further path than Pure pursuit (GPS) which can be driven in areas without lanes.

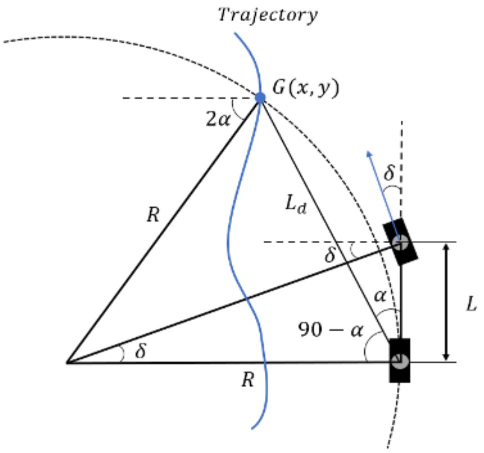


FIGURE 4. Geometric explanation of Pure pursuit.

Algorithm 2 Steering angle compensation algorithm

```
Input: $\delta$: Steering angle calculated with Pure pursuit and filter
Output: $com\_\delta$: Steering angle after compensation
$T$: Threshold
if $\delta$ - $prev\_steer > T$ then
    $com\_\delta = prev\_steer + T$
else if $prev\_steer$ - $\delta > T$ then
    $com\_\delta = prev\_steer - T$
else
    $com\_\delta = \delta$
end if
$prev\_steer = com\_\delta$
return $com\_\delta$
```

2) Stanley Controller

Like the Pure pursuit, the Stanley controller calculates the steering angle based on the bicycle model. Two types of errors are used to calculate the steering angle. The two errors consist of a cross-track error and a heading error. The cross-track error refers to the minimum distance between a given path and a front wheel of the vehicle. The heading error refers to the difference between the direction vector of the path and the direction vector of the vehicle. Using these two errors, the steering angle can be obtained and a geometric explanation for this algorithm is shown in Fig. 5.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

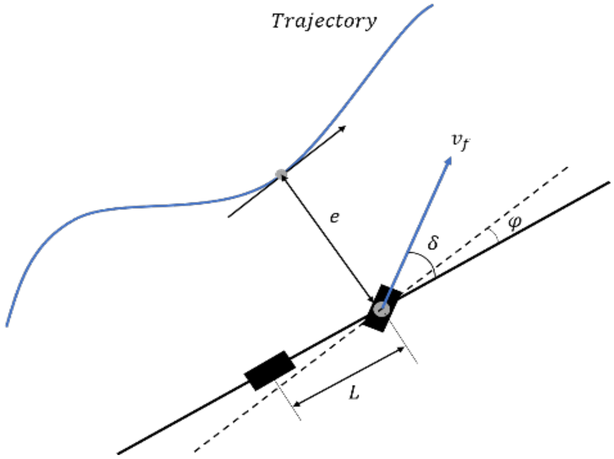


FIGURE 5. Geometric explanation of Stanley controller.

$$\delta(t) = \theta_e(t) + \theta_d(t) = \varphi(t) + \tan^{-1}\frac{ke(t))}{k_s + v_f(t)} \tag{4}$$

where $\varphi(t)(\theta_e(t))$ is heading error, $e(t)$ is cross-track error, $v_f(t)$ is the speed of the vehicle, k and k_s are gain parameters respectively. k and k_s in parameter types are expressed in (4). Since it is difficult to implement perfect tracking performance with only these two parameters, k_1 and k_2 should be added before $\theta_e(t)$ and $\theta_d(t)$, respectively. Therefore, the modified Stanley controller expression can be written the same as (5).

$$\delta(t) = k_1\theta_e(t) + k_2\theta_d(t) = k_1\varphi(t) + k_2\tan^{-1}\frac{ke(t))}{k_s + v_f(t)} \tag{5}$$

This modified Stanley controller was used to follow the global path provided by the HD map. Stanley (GPS) and Stanley (vision) represent Stanley controller using GPS based path and vision based path respectively. Stanley (GPS) receives the current position of the vehicle through GPS and the position of the front wheel of the vehicle can be calculated using L, the wheel base of the vehicle. In case of using the modified Stanley (GPS), the vehicle shakes a lot. Nevertheless, Stanley (GPS) is necessary because it can be driven in very large curvatures.

Stanley controller does not use the vision based path due to the method of determining the steering angle. As described above, Stanley controller uses the heading error and the cross-track error to calculate the steering angle. Since the vision based path starts from the front wheel of the vehicle, the cross-track error is not able to exist as zero. With Stanley (vision), the steering angle must be calculated only from the heading error. Therefore, Stanley (vision) was not used because of its low tracking performance.

D. PATH PROCESSING METHOD

The cameras, GPS and HD map were used to generate paths. GPS and HD map are essential to get the path, when lane detection was not able to properly perform or in the case of large curvature road. It was explained in the Section III-B that the path could be retrieved from raw image through the camera. The process of creating a vision based path included a process in which the tracker uses the path immediately. In order to use the GPS based path directly in the tracker like the vision based path, coordinate transformation is required. The process of smoothly connecting paths becomes also necessary for more stable driving. Therefore, path processing method includes the coordinate transformation and interpolation.

1) Path Processing Method based on Vision (Pure Pursuit)

The processing method of path based on vision used in Pure pursuit will be explained as follows. The points obtained in section III-B are assigned a corresponding distance estimated from the camera extrinsic parameter. The look-ahead distance varies with velocity of vehicle and the look-ahead point also changes. The target point, the closest point to the look-ahead point, affects the steering angle. Since the points are not continuous, the target point changes from time to time, which leads to instability in steering. In order to ensure that the target point is located as close to the look-ahead point as possible, the number of points was increased by selecting the two closest points to the look-ahead point and using linear interpolation between the two points. The stability of the steering angle is compensated by tracking the target point at the same location as the look-ahead point.

2) Path Processing Method based on GPS (Stanley Controller)

The processing method of global path used in Stanley controller is shown in Fig. 6. First of all, the yaw value of the vehicle is obtained using IMU. The global path is received from the HD map and provides 30 points with an interval of about 1 meter based on the current location of the vehicle. In order to use this path in Stanley controller, it is necessary to connect 30 points smoothly in a curve. Therefore, cubic spline interpolation was used to approximate these points as curves. Cubic spline interpolation represents a method of smoothly connecting given points using a cubic polynomial. By applying this, 30 points that come in at intervals of 1 meter are interpolated at intervals of about 0.1 meter, and all yaw values corresponding to each point are calculated. Using the interpolated path and vehicle's yaw value, the distance to the point on the path closest to the current vehicle and yaw at which point can be calculated. Using these values, the steering angle is finally obtained by the Stanley controller.

3) Path Processing Method based on GPS (Pure Pursuit)

The processing method of the global path used in Pure pursuit provides 30 points at about 1 meter intervals based on the current location. The coordinate system of the global

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

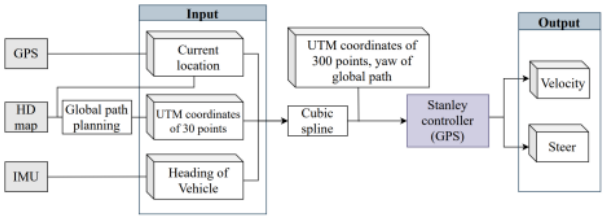


FIGURE 6. Path processing method based on GPS used in Stanley controller.

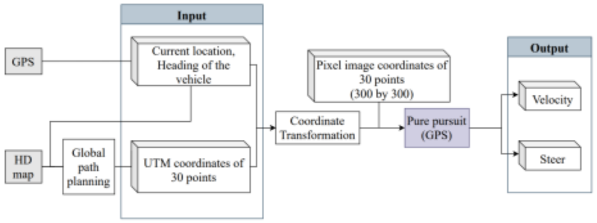
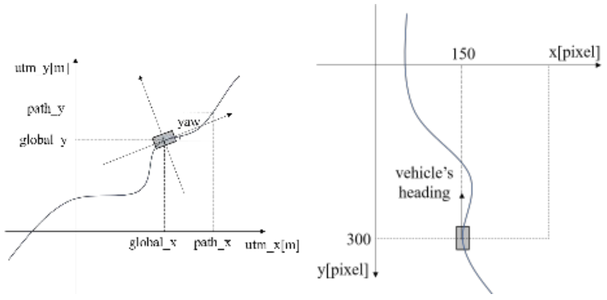

Input
Output
GPS
HD map
Current location, Heading of the vehicle
Global path planning
UTM coordinates of 30 points
Pixel image coordinates of 30 points (300 by 300)
Coordinate Transformation
Pure pursuit (GPS)
Velocity
Steer

FIGURE 8. Path processing method based on GPS used in Pure pursuit.

path uses the UTM coordinate system and the coordinate system used in the Pure pursuit requires local path. Therefore, the heading of the vehicle was always set to be the same as the direction of the y-axis, and coordinate transformation converting the global path into a local path was used. Since 300 by 300 pixel image coordinate was used, scaling was performed to fit the units of the coordinate system. These path coordinate transformations are shown in Fig.7, and the coordinate transformation equations are formulated in (6) and (7). The data-flow diagram for the path processing method based on GPS used in Pure pursuit is shown in Fig. 8.

$$new\_path\_x = 150 - 10 \cdot [cos\{-(90+yaw)\} \cdot (path\_x - global\_x) - sin\{-(90+yaw)\} \cdot (path\_y - global\_y)] \tag{6}$$

$$new\_path\_y = 300 + 10 \cdot [cos\{-(90+yaw)\} \cdot (path\_x - global\_x) - sin\{-(90+yaw)\} \cdot (path\_y - global\_y)] \tag{7}$$



(a) (b)

FIGURE 7. Path coordinate transformation (a) UTM coordinate system, (b) pixel image coordinate system.

E. OPTIMAL PATH TRACKER SELECTION

The most important part of hybrid tracker based optimal path tracking system becomes optimal path tracker selection. Optimal path tracker selection chooses an appropriate path tracking algorithm among Pure pursuit (GPS), Pure pursuit (vision), and Stanley (GPS) considering instability of steering angle, GPS reliability, vision reliability, and road information. The selected trackers can be changed in short intervals alternately because this algorithm ensures real-time and selects the optimized tracker. The instability of steering angle is predetermined based on the driving tendency of the Stanley controller and Pure pursuit. GPS determines the reliability of the sensor using horizontal dilution of precision (HDOP) and RTK precision information. Vision checks the information of the path received from section III-B and determines the detection reliability of the lane itself inside the system. It also determines road information such as curvature, roundabout, and intersection of the path to be driven based on HD map. The key role in the optimal path tracker selection is to observe the information provided by the sensors and to switch securely the path tracker.

The instability of the steering angle is determined according to the following results shown in Fig. 9. Comparing the steering values, the vibration of the steering angle using Pure pursuit becomes smaller than using the Stanley controller. When Pure pursuit was used, it converged to 0 degrees in 4 seconds, whereas when Stanley was used, it oscillated within 10 degrees and -10 degrees even after 4 seconds. It can be deduced that the driving stability would be better if the Pure pursuit with higher stability of steering angle was used in the range of driving with in-track maintained.

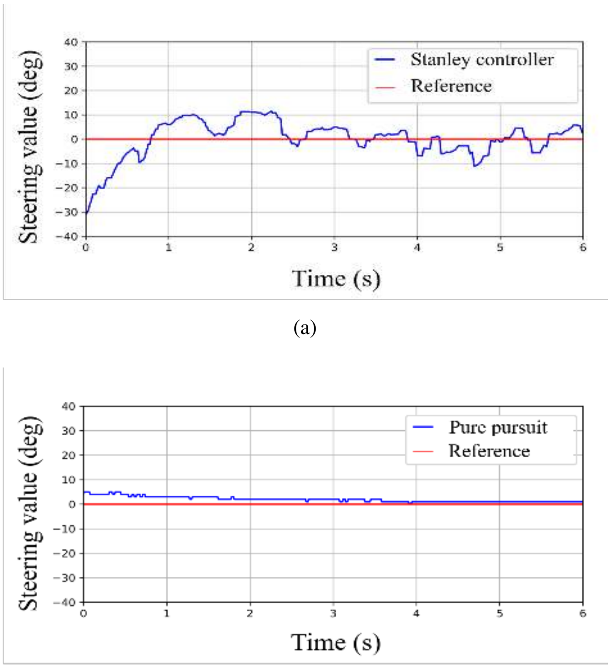
In this research, GPS state information is named as GPS reliability, HD map based road information is named as HD map data and vision based lane detection state information is named as lane reliability. Each evaluating method is explained as follows.

First, HDOP refers to the degree of interfering with the positional precision of the horizontal coordinate by parsing from the information of NMEA 0183 of GPS. Therefore, HDOP and RTK precision information was obtained from GPS, and GPS reliability was evaluated by combining them. The normal operating standard of HDOP was set to 3 or less experimentally. The information of RTK precision is divided into 3 levels, which the highest level (Fixed) was treated as 2, the lower levels in turn were treated as 1 (Float) and 0 (No Fixed). The Fixed option was chosen in this research as the highest level because GPS only works well with Fixed, which compensates for signals using LTE. If HDOP is less than 3 and the RTK precision information satisfies Fixed option, the GPS determines only that it is operating normally as shown in TABLE 2.

The method of evaluating the lane reliability is explained

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access



(a)

(b)

FIGURE 9. Steering at straight road (a) steering angle using Pure pursuit, (b) steering angle using Stanley controller.

TABLE 2. Criteria on determination of the GPS availability.

RTK precision information	HDOP	GPS reliability
No Fixed(0)	$HDOP < 3$	×(0)
	$HDOP \geq 3$	×(0)
Float(1)	$HDOP < 3$	×(0)
	$HDOP \geq 3$	×(0)
Fixed(2)	$HDOP < 3$	○(1)
	$HDOP \geq 3$	×(0)

as follows. In the 300 by 300 window, the location of the current vehicle was set to (150,0) and the vision based path in the direction of heading was provided as point group information. By expressing the location and point group information as an image, it was possible to determine the pixel distance as metrics for the relationship between the path and the current vehicle location.

The pixel coordinates of the x-axis were scaled at 0.03 meter per pixel. The width of a road equals 100 pixels. Considering width of the vehicle, if the vehicle is off one-fifth the width of the road from its current location, it could be deduced that the test vehicle invaded the center line. Therefore, if the Equation (8) is satisfied, the vision based path is determined to be stable and 1 is provided as lane reliability.

$$|150 - lpts| < W_r/5 \tag{8}$$

where $lpts$ is x coordinates of lane points and W_r is width of a road in the pixel coordinates respectively.

When lanes are unclear in the intersection and the roundabout, it is difficult for vision to recognize them. In tunnel or a road on a hillside, GPS status is not good, which leads GPS based path to a low reliability. Therefore, the driving stability should be improved by receiving information of road terrain in advance and selecting the optimal path tracking algorithm.

By using the optimal path tracker selection with these information, each of the disadvantages was compensated and solved simply by switching the tracking algorithm following each path of vision and GPS.

The criteria of this system for selecting the tracking algorithm using the data described above is summarized in TABLE 3.

TABLE 3. Criteria of optimal path tracker selection.

GPS reliability	HD map data	Lane reliability	Selected path tracker
0	*	*	Pure pursuit (vision)
1	1	1	Stanley controller (GPS)
1	1	0	Stanley controller (GPS)
1	0	1	Pure pursuit (vision, GPS)
1	0	0	Pure pursuit (vision, GPS)

’*’ means that it doesn’t matter if there is any value (0 or 1).

Each part from perception to control was described in detail above. In order to reduce the computational cost, the perception and control sections were closely connected, and the trackers were configured as parallel systems. An overview of the architecture of Hybrid tracker based optimal path tracking system used in this research is shown in Fig. 10.

IV. EXPERIMENTAL RESULTS

This section includes the performance analysis about the proposed Hybrid tracker based optimal path tracking system (hereinafter referred to as Hybrid). All of the tested algorithms were mounted on Hyundai Ioniq Electric 2020 and were tested on a shuttle route of DGIST and on high-speed circuit proving ground. Tested roads comprise a variety of roads, including two-lane or four-lane roads, intersection, as well as steep curve, tunnels, roundabout and so on. In addition, the shuttle route includes both uphill, flat and downhill, enabling acceleration testing. The high-speed circuit consists of three-lane one-way and banked curve whose radius is 100 meters. The roads in DGIST can be assumed as Korean urban road and the high-speed circuit might be assumed highways. Therefore, tested paths were considered good road conditions for verifying the system at low and high speeds (*i.e.*, 0 km/h to 100 km/h).

The DGIST shuttle route was analyzed into normal road environments and complex road environments. Roadways relatively easy to follow among the DGIST shuttle route are called the normal road environment. The three types of

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

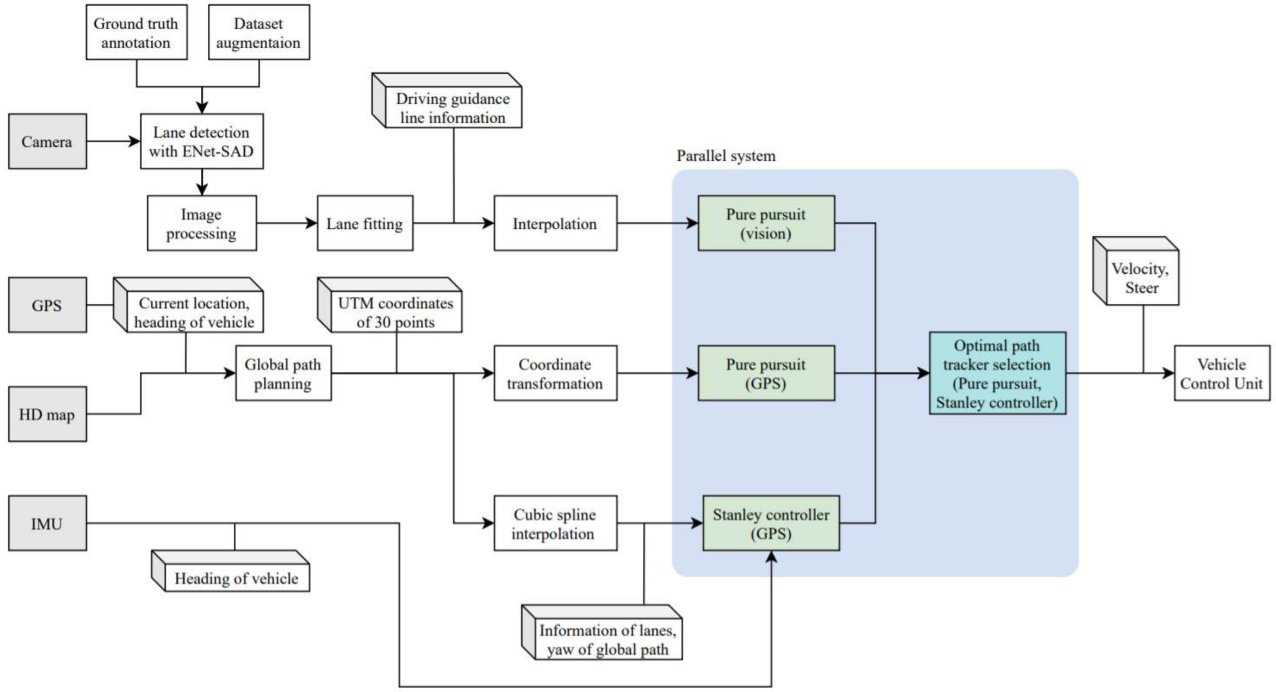

Ground truth annotation
Dataset augmentaion
Camera
Lane detection with ENet-SAD
Driving guidance line information
Parallel system
Image processing
Lane fitting
Interpolation
Pure pursuit (vision)
Velocity, Steer
GPS
Current location, heading of vehicle
UTM coordinates of 30 points
Global path planning
Coordinate transformation
Pure pursuit (GPS)
Optimal path tracker selection (Pure pursuit, Stanley controller)
Vehicle Control Unit
HD map
IMU
Cubic spline interpolation
Stanley controller (GPS)
Heading of vehicle
Information of lanes, yaw of global path

FIGURE 10. General overview of the Hybrid tracker based optimal path tracking system.

normal road environment were selected as straight, slight curve of radius 157.5 meters and steep curve of radius 53 meters because tracker is able to drive properly on these roadways using only one of the vision and GPS based path.

Sections of the road relatively difficult to follow among the DGIST shuttle route are called the complex road environment, and the four sections were designated as straight road with multiple 3-way junctions, roundabout of radius 14 meters, tunnel, and intersection. The reasons why the sections were selected as the complex road environment will be explained as follows.

Unlike roads belonging to the normal road environment, the intersection and roundabout have very large curvature and the cameras cannot recognize the lanes in the road of large curvature. In addition, since lanes occasionally disappears for straight roads with multiple 3-way junctions, GPS based path is required. Due to the low reliability of GPS in the tunnel, the vision based path is only available.

A. PERFORMANCE OF FAST OPTIMAL LANE PROCESSING ALGORITHM

The performance of fast optimal lane processing algorithm will be explained in this section. The Fig. 11 shows the image overlapping the raw image with the lane of top view image processed by the fast optimal lane processing algorithm. In the Fig. 11, (a), (b), (c) and (d) show the results of lane processing algorithm at straight road, roads with lane covered by vehicles, curve road, tunnel respectively. The results of straight roads consist of uphill, flat and downhill roads. The results of the tunnel show that the vehicle passes through the tunnel. The algorithm recognizes lanes despite of structures or vehicles on the road, such as speed bump, eye-inducing rods, vehicles. It can be deduced that this algorithm robustly recognizes lanes in various road environments and find the best fitting regression function.

The time-averaged to infer lanes from ENet-SAD is taken 23.53ms. The time-averaged to fit the pixel coordinates is measured 4.42ms. The sum of several delays is approximately found 5ms. Therefore, the final runtime is 32.95ms.

B. EVALUATION ON REAL-WORLD

The modified Pure pursuit, Stanley controller and Hybrid system have been tested and analyzed in complex road conditions. Two types of driving guidance line, the vision based path and the GPS based path can be used by each path tracking algorithm. There are four possible cases; Pure pursuit (vision), Pure pursuit (GPS), Stanley (vision), and Stanley (GPS). As for these four cases, the other three cases excluding Stanley (vision) which had poor performance have been studied in this research. The detailed reason for excluding the Stanley (vision) was explained in section III-C2.

Therefore, the three modified trackers were optimized for coordinate transformation and post-processing algorithms of various sensor parts. Tracking performance and driving stability have been evaluated by comparing the Hybrid and three modified trackers with ideal path based on HD map. *Success* represents that the tracking algorithms drove in the designated area without hitting the curb or moving into the opposite lane

To determine the tracking performance of each tracker, RMSE (*lateral*), RMSE (*longitudinal*) and *distance* were used in metric unit. These values represent the difference

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

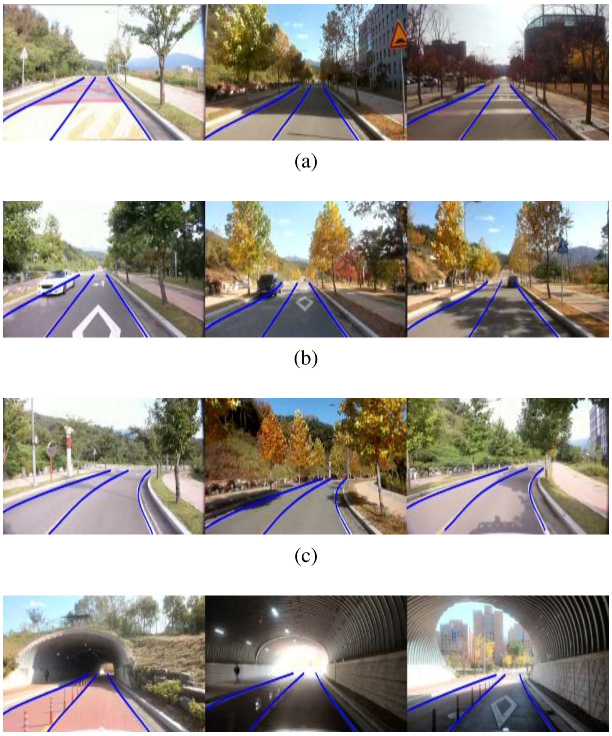


FIGURE 11. Estimation results after applying the fast optimal lane processing algorithm with diverse road conditions (a) straight roads, (b) roads with lane covered, (c) curve roads, (d) tunnel.

between the ideal path and the real driving path. RMSE (*lateral*) and RMSE (*longitudinal*) stand for the root mean square error of latitude and longitude, respectively. The "*distance*" signifies the difference between the ideal path and the real driving path.

Driving stability has been determined using the RMSE (*yaw*) and RMSE (*steer*) in degree unit. RMSE (*yaw*) expresses the difference between the yaw of the vehicle and the yaw of HD map. RMSE (*steer*) means the difference between the steering angle directly driven by a human in an ideal path and the steering angle calculated by each path tracking algorithm. The *yaw* indicates vehicle's heading, and the *steer* means a value that determines the direction of the vehicle. The small RMSE (*yaw*) value represents that the planned driving direction becomes similar to the ideal driving direction. The small RMSE (*steer*) value also implies that the steering value calculated by the tracking algorithm becomes similar to steer of the human, since it is compared to the driving smoothly of human. Therefore, if both values show small, it implies driving stability is high because the algorithm follows the path smoothly without fluctuation. Thus, the driving stability can be evaluated with values of RMSE (*yaw*) and RMSE (*steer*).

These six experimental results were summarized in TABLE 4 and 5 to find easily out the differences, and the Top-2 results for each column was emphasized in these tables.

1) Test Results Under Normal Road Environment

Fig. 12 shows graphically the tracking results of Pure pursuit (vision), Pure pursuit (GPS), Stanley (GPS) and Hybrid in three different scenes. In Fig. 12(a), (b) and (d) shows the results; a straight road, a slight curve road, and a steep curve road. Graph (c) and (e) are enlarged views of the yellow boxes in graphs (b) and (d) in Fig. 12, respectively. It can be seen in the figure that Pure pursuit (vision), Pure pursuit (GPS), Stanley (GPS), and Hybrid are all almost identical to the ideal path in the normal road environment. Numerical interpretation of the graph is demonstrated in TABLE 4.

Thus, it can be concluded that all of modified trackers successfully drove the suggested section without infringement on the lane from the *Success* column in TABLE 4). As mentioned in section III-C in this research, each tracker clearly shows high performance in the certain section of the road due to the characteristics of each tracker. Therefore, this implies that the result of modified each tracker produces optimal performance for each situation. Hybrid system shows intermediate or higher results in the overall result. Since the Hybrid system operates by selecting several optimal trackers at a fast computational speed, it is possible to secure at least the average performance of each tracker.

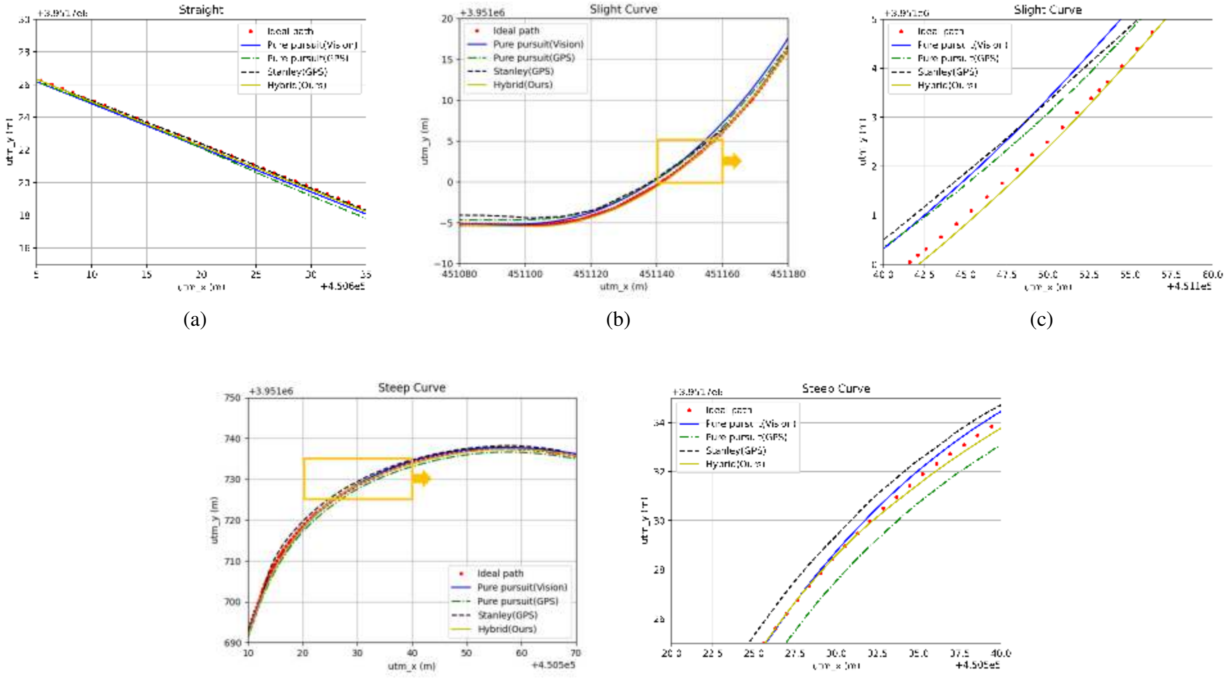
In the case of straight road, the overall tracking performance (*i.e.*, RMSE) of Hybrid system is lower than the peak performance of other base trackers. It is because Hybrid is the system composed of the above three trackers. The Hybrid system also guarantees of both stability and tracking performance, whereas these slight losses may occur. As shown in TABLE 4 and 5, in terms of stability, the Hybrid achieves the highest performance. RMSE (*yaw*) and RMSE (*steer*) values were lower than the second best performances of Pure pursuit (vision) by 0.054 and 0.014 in degree unit respectively. Thus, it convinces that Hybrid is superior in terms of driving stability. It is also found that Hybrid has a tendency to lack the ability to compensate for residual errors in a straight road situation as it is changed to multiple trackers due to the fast selection speed. However, due to these characteristics, robustness of the stability for the Hybrid system improved the overall driving performance.

The driving performances show the most effective in the slight curve due to the characteristics of Hybrid. It is also able to be found that other roads have a clear difference in performance between vision and GPS, but slight curve shows relatively similar performance. In other words, the Hybrid has the best overall performance in slight curve because the tracking performance of multiple trackers is properly used. Conversely, RMSE (*yaw*) leads to 16.933 degrees out of ideal path, which is the third best among the four algorithms. However, the difference between the value of the best algorithm and Hybrid was only 0.686 degrees which equals to only 0.04% differences with the results of the other algorithms. Since the value of RMSE (*steer*) was the smallest one, it can be explained that the driving stability of Hybrid system provides the best comfortableness.

Moreover, in the steep curve, due to the limitations of

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access



(a) (b) (c)

(d) (e)

FIGURE 12. Test result at normal road environments (a) straight road, (b) slight curve, (c) enlarged graph of (b), (d) steep curve, (e) enlarged graph of (d).

TABLE 4. Quantitative results by different path tracking algorithms under normal road environments (*i.e.*, straight, slight curve, steep curve).

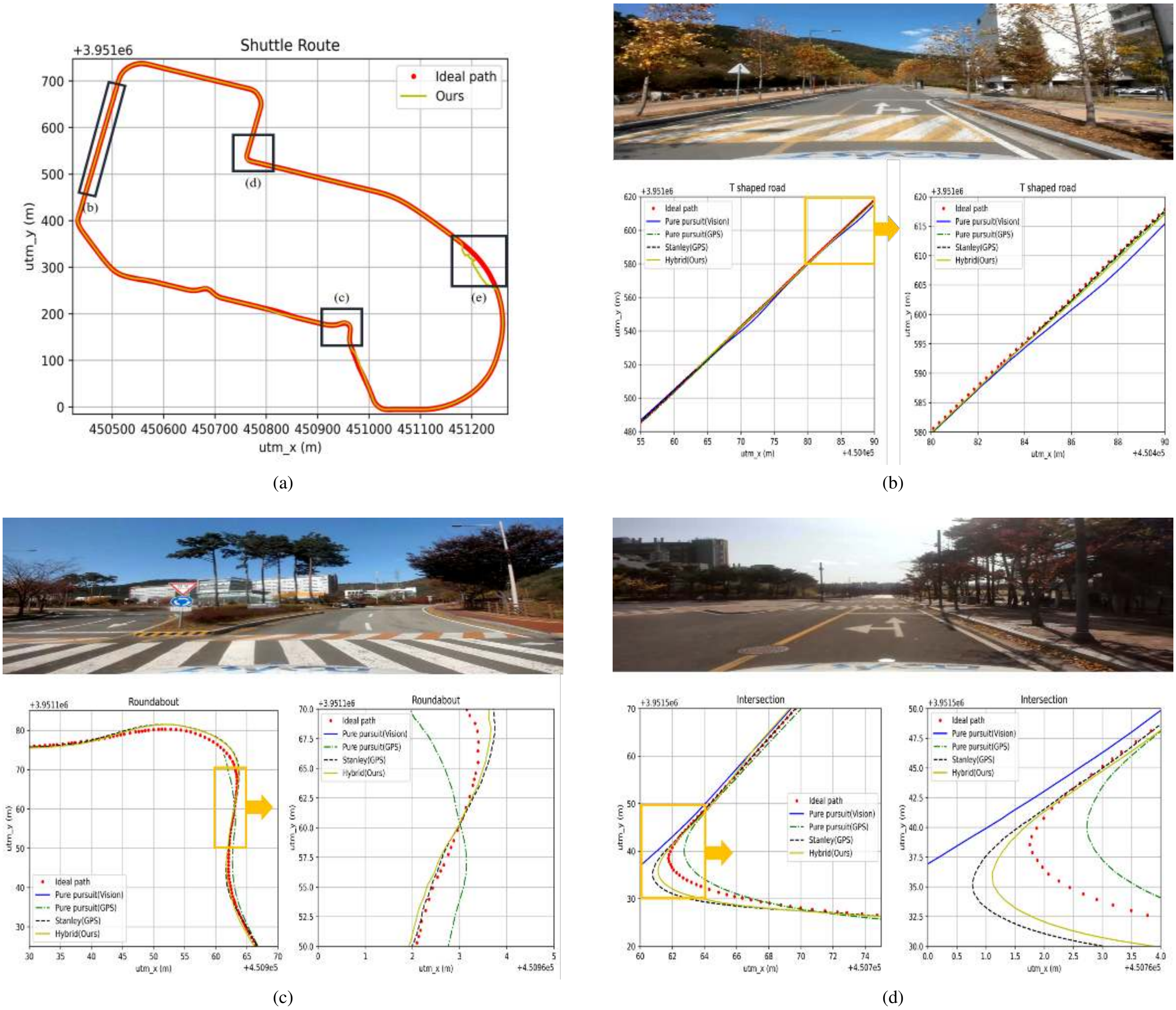
Scenes	Methods	*Success*	Tracking Performance			Driving Stability	
			RMSE (*lateral*) ↓	RMSE (*longitudinal*) ↓	*Distance* ↓	RMSE (*yaw*) ↓	RMSE (*steer*) ↓
Straight	Pure pursuit (vision)	◯	**0.091**	**0.019**	**0.068**	**0.079**	**1.314**
	Pure pursuit (GPS)	◯	0.223	**0.007**	0.155	0.174	1.755
	Stanley (GPS)	◯	**0.196**	0.065	**0.146**	0.158	2.406
	Hybrid	◯	0.377	0.049	0.285	**0.025**	**1.300**
Slight curve	Pure pursuit (vision)	◯	**0.027**	**0.283**	**0.207**	17.742	**2.270**
	Pure pursuit (GPS)	◯	0.239	0.577	0.598	**16.685**	2.894
	Stanley (GPS)	◯	0.131	0.553	0.463	**16.247**	3.430
	Hybrid	◯	**0.027**	**0.169**	**0.144**	16.933	**1.929**
Steep curve	Pure pursuit (vision)	◯	0.450	0.397	0.522	**57.994**	**2.865**
	Pure pursuit (GPS)	◯	**0.129**	0.492	**0.384**	62.959	**3.199**
	Stanley (GPS)	◯	**0.145**	**0.174**	**0.166**	63.501	5.406
	Hybrid	◯	0.321	**0.370**	0.414	**52.047**	3.485

The unit of RMSE ($lateral$), RMSE ($longitudinal$) and $Distance$ is meter. The unit of RMSE (yaw) and RMSE ($steer$) is degree.

vision, trackers using GPS are mainly used. In this case, Stanley (GPS) shows high precision for this road environment. However, due to bad stability, Hybrid uses only Pure pursuit (vision) and Pure pursuit (GPS) to secure overall driving performance. As an experimental result, the tendency of error becomes similar that of Pure pursuit (vision). The tracking performance of Hybrid gets worse than the performance of Stanley (GPS) since Hybrid uses Pure pursuit trackers. Thus, it can be deduced that relatively low tracking performance of Hybrid system is a result of the algorithm operation to ensure stability. When human drives on the steep curve, they usually do offset driving for driving stability. Since the proposed system does not have environment recognition, it is important to precisely follow the provided path. Hence, when the importance of the in-track is high, such as the steep curve, the Hybrid drives by focusing on the in-track rather than the stability. In the steep curve, the results show Stanley (GPS) has the best tracking performance, but show Stanley (GPS) has poor driving stability. Therefore, it can be concluded that Hybrid secured driving stability by properly switching two trackers, Pure pursuit (vision) and Pure pursuit (GPS), which have good stability performance in the secured tracking performance situation. Considering the value of RMSE ($steer$) compared to humans with offset driving, the smaller value of RMSE (yaw) than 5 degrees compared to other trackers imply that the overall driving performance of Hybrid in steep curve represents superior.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access



(a)

(b)

(c)

(d)

FIGURE 13. DGIST shuttle route and experimental results at complex road environments (a) DGIST shuttle route, (b) straight road with multiple 3-way junctions, (c) roundabout, (d) intersection.

2) Test Results Under Complex Road Environment

In the previous section, it was found that three tracking algorithms and Hybrid system succeeded in driving the normal road environment. However, there were some sections where three algorithms except Hybrid could not properly control the vehicle in roundabout, intersection, and tunnel shown in the Fig. 13. The Fig. 13(a) shows that the path in which the proposed system actually drove the entire DGIST shuttle route closely matches the ideal path. Each road in the complex road environment is marked by a black box in Fig. 13(a). Figures of (b), (c) and (d) comprise a combination of the roadway actual tested, the driving graph and the graph extending the yellow box. Fig. 13(b) shows the results of tracking straight roads with multiple 3-way junctions and the enlarged graph demonstrates that Pure pursuit (vision) has poor tracking performance due to the absence of lanes in the area with 3-way junctions. Fig. 13(c) displays the results of tracking the roundabout. Pure pursuit (vision) was unable to follow because the lane of the roundabout was not clear and Pure pursuit (GPS) was unable to follow accurately at very large curvature. Only Stanley controller succeeded in tracking and the proposed system tracked well by selecting Stanley controller among the three tracking algorithms. Fig. 13(d) shows the results of tracking the intersection. Pure pursuit (vision) failed to follow the driving guidance line of the intersection. Pure pursuit (GPS) and Stanley controller tracked the path well. Pure pursuit (GPS) adjusted the steers quickly and Stanley controller adjusted the steers slowly, whereas Pure pursuit (GPS) occasionally crossed the center line and tracked the path overall.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

TABLE 5. Quantitative results by different path tracking algorithms under complex road environments (*i.e.*, straight road with multiple 3-way junctions, roundabout, tunnel, intersection).

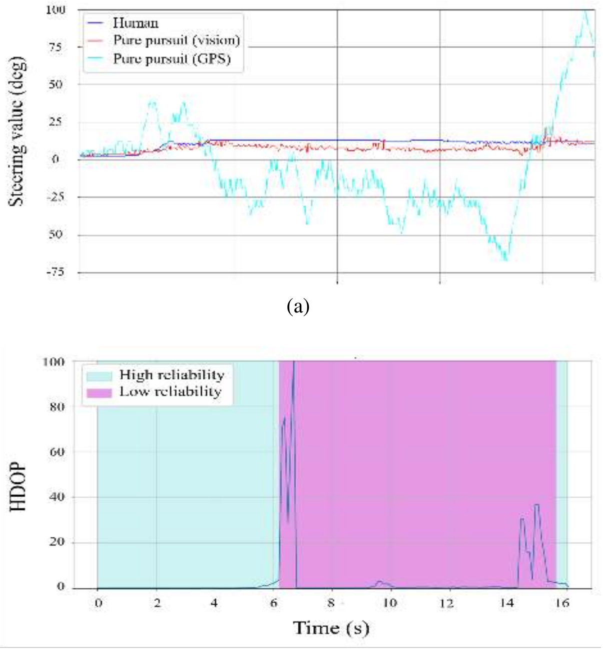
Scenes	Methods	*Success*	Tracking Performance			Driving Stability	
			RMSE (*lateral*) ↓	RMSE (*longitudinal*) ↓	*Distance* ↓	RMSE (*yaw*) ↓	RMSE (*steer*) ↓
Straight road with multiple 3-way junctions	Pure pursuit (vision)	○	0.152	1.063	0.862	0.461	**1.671**
	Pure pursuit (GPS)	○	**0.029**	0.290	0.219	2.871	2.434
	Stanley (GPS)	○	0.081	**0.183**	**0.142**	**0.125**	2.734
	Hybrid	○	**0.056**	**0.214**	**0.157**	**0.018**	**1.660**
Roundabout	Pure pursuit (vision)	×	-	-	-	-	8.252
	Pure pursuit (GPS)	×	-	-	-	-	6.431
	Stanley (GPS)	○	0.415	**0.403**	0.471	187.166	**5.912**
	Hybrid	○	**0.232**	0.486	**0.377**	**172.815**	**5.708**
Intersection	Pure pursuit (vision)	×	-	-	-	-	6.664
	Pure pursuit (GPS)	○	**0.130**	0.263	**0.227**	75.829	6.046
	Stanley (GPS)	○	**0.424**	**0.109**	**0.260**	**73.313**	**5.493**
	Hybrid	○	0.439	**0.181**	0.330	**65.093**	**5.481**
Tunnel	Pure pursuit (vision)	○	-	-	-	-	**1.935**
	Pure pursuit (GPS)	×	-	-	-	-	7.704
	Stanley (GPS)	×	-	-	-	-	8.305
	Hybrid	○	-	-	-	-	**1.911**

The unit of RMSE (*lateral*), RMSE (*longitudinal*) and *Distance* is meter. The unit of RMSE (*yaw*) and RMSE (*steer*) is degree.

In the complex road environment, a numerical analysis of the tracking performance and driving stability of each algorithm are listed in TABLE 5. In straight road with multiple 3-way junctions, the algorithms using the GPS based path have better tracking performance than the algorithm using the vision based path since lanes are often disconnected. The *distance* value of Hybrid is 0.157 meter, which can be seen to have a value between the *distance* value of Stanley (GPS) and the *distance* value of Pure pursuit (GPS). It might be implied that the Hybrid selected alternately Stanley (GPS) and Pure pursuit (GPS). In particular, the values of RMSE (*yaw*) and RMSE (*steer*) representing a driving stability showed the best among the tested methods.

In roundabout, due to the few lanes, Pure pursuit (vision) was unable to drive properly this road. Also, it was unable to adopt Pure pursuit (GPS) due to very large curvature. The TABLE 5 also indicated that Stanley (GPS) and Hybrid succeeded in driving the roundabout, and *distance* value of Hybrid is about 0.094 meter closer than that of Stanley (GPS)'s *distance* to the ideal path. In terms of driving stability, RMSE (*yaw*) value of Hybrid showed 14.351 degrees better than that of Stanley (GPS), and RMSE (*steer*) of Hybrid is slight difference with that of Stanley (GPS). This is the reason why there is a big difference (*i.e.*, 14.3 degrees) in RMSE (*yaw*) even though Hybrid adopted Stanley (GPS) because Hybrid chose different trackers in entering the roundabout. It can be finally induced that Hybrid becomes superior in terms of tracking performance and driving stability.

In intersection, Pure pursuit (GPS), Stanley (GPS), Hybrid system succeeded. The *distance* values show all 0.330 meter or less, so the tracking performance indicates high. Although Hybrid did not represent the highest tracking performance among them, it guarantees the driving stability.



(a)

(b)

FIGURE 14. Experimental results in tunnel (a) comparison of steering angle value, (b) GPS reliability.

In Tunnel, the GPS did not work properly, so tracking performance could not be obtained. The tracking of the tunnel area was operated well based on Pure pursuit (vision), but the reliability of the GPS showed low and the graph seems that the vehicle did not follow the path, as in the (e) area of Fig. 13(a). It is shown that the steer value driven by humans and the steer values predicted by each algorithm. Fig. 14(b) shows reliability of GPS. When the reliability of GPS becomes high, the graph background color is highlighted in cyan, otherwise in magenta. The steer value provided by Pure pursuit (vision) shows pretty similar to the steer value driven by humans, whereas the steer value provided by Pure pursuit (GPS) does not. It is also shown in Fig. 14(a) that algorithms

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access

using GPS based path are not able to pass through tunnels due to the low reliability of GPS.

Finally, since GPS based path was impossible to operate correctly in the tunnel, only RMSE (steer) was obtained. RMSE (*steer*) of Pure pursuit (vision) and RMSE (*steer*) of Hybrid are similar with 0.6 degrees differences. Therefore, Hybrid succeeded in passing through the tunnel using Pure pursuit (vision). In summary of experimental results, the path tracking algorithms for the vision based path and the GPS based path were appropriately switched through the Hybrid tracker based optimal path tracking system to drive the shuttle route on DGIST campus.

V. CONCLUSION

This research proposes Hybrid tracker based optimal path tracking system for autonomous vehicles. The designed system combines, evaluates and applies several trackers to driving on complex road conditions. The proposed system includes all of the processing algorithms of deep learning based lane detection algorithms, coordinate system transformation, three modified geometric trackers and optimal path selection algorithm. Based on combining all above processes, the proposed system guarantees reasonable driving performance for trade-off of driving stability and tracking performance in both normal and complex road environments. This study also notes that the quantitative comparison using ideal path and the steering value from human shows the overall driving performance of Hybrid tracker based optimal path tracking system. With experimental results, this study convinces to build a seamless system from the sensor part of the autonomous vehicle to the action part for better performance. In other words, the key to build this system was that control and sensor recognition should be made in one process. The proposed system secured improved driving stability and tracking performance with low computation cost by applying the hybrid method to the tracker using GPS and vision. However, since it is a method of improving at the tracking stage, not path planning, it can show high stability only in HD map based driving (*i.e.*, Shuttle and perfectly planned route) provided in advance. It is considered that stable results can be derived with path planning using HD map or long-distance vision path. Consequently, as shown in experimental results, it notes that proposed system of the optimal path selection and system architecture significantly improves both driving stability and tracking performance in the presence of high complexity of road conditions.

REFERENCES

[1] S. O.-R. A. V. S. Committee *et al.*, “Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems.(2014),” 2014.

[2] B. Lu, H. He, H. Yu, H. Wang, G. Li, M. Shi, and D. Cao, “Hybrid path planning combining potential field with sigmoid curve for autonomous driving,” *Sensors*, vol. 20, no. 24, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/24/7197

[3] K. Lee, S. Jeon, H. Kim, and D. Kum, “Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic gaussian (lqg) control,” *IEEE Access*, vol. 7, pp. 109 120–109 133, 2019.

[4] L.-A. Tran and M.-H. Le, “Robust u-net-based road lane markings detection for autonomous driving,” in *2019 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2019, pp. 62–66.

[5] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, “Towards end-to-end lane detection: an instance segmentation approach,” in *2018 IEEE intelligent vehicles symposium (IV)*. IEEE, 2018, pp. 286–291.

[6] Y. Xing, C. Lv, L. Chen, H. Wang, H. Wang, D. Cao, E. Velenis, and F.-Y. Wang, “Advances in vision-based lane detection: algorithms, integration, assessment, and perspectives on acp-based parallel vision,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 645–661, 2018.

[7] Q. Khan, T. Schön, and P. Wenzel, “Latent space reinforcement learning for steering angle prediction,” *arXiv preprint arXiv:1902.03765*, 2019.

[8] B. Lu, G. Li, H. Yu, H. Wang, J. Guo, D. Cao, and H. He, “Adaptive potential field-based path planning for complex autonomous driving scenarios,” *IEEE Access*, vol. 8, pp. 225 294–225 305, 2020.

[9] J. M. Snider *et al.*, “Automatic steering methods for autonomous automobile path tracking,” *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.

[10] M. Cibooglu, U. Karapinar, and M. T. Söylemez, “Hybrid controller approach for an autonomous ground vehicle path tracking problem,” in *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2017, pp. 583–588.

[11] Q. Yao, Y. Tian, Q. Wang, and S. Wang, “Control strategies on path tracking for autonomous vehicle: State of the art and future challenges,” *IEEE Access*, vol. 8, pp. 161 211–161 222, 2020.

[12] M. Kim, G. Im, and J. Park, “A comparative study of parking path following methods for autonomous parking system,” *The Journal of Korea Robotics Society*, vol. 15, no. 2, pp. 147–159, 2020.

[13] N. H. Amer, K. Hudha, H. Zamzuri, V. R. Aparow, A. F. Z. Abidin, Z. Abd Kadir, and M. Murrad, “Adaptive modified stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle,” *Robotics and Autonomous Systems*, vol. 105, pp. 94–111, 2018.

[14] M.-W. Park, S.-W. Lee, and W.-Y. Han, “Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm,” in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*. IEEE, 2014, pp. 1443–1447.

[15] S. Dominguez, A. Ali, G. Garcia, and P. Martinet, “Comparison of lateral controllers for autonomous vehicle: Experimental results,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1418–1423.

[16] Z. Chen, Q. Liu, and C. Lian, “Pointlanenet: Efficient end-to-end cnns for accurate real-time lane detection,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2563–2568.

[17] Y. Ko, J. Jun, D. Ko, and M. Jeon, “Key points estimation and point instance segmentation approach for lane detection,” *arXiv preprint arXiv:2002.06604*, 2020.

[18] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, “Learning lightweight lane detection cnns by self attention distillation,” *arXiv preprint arXiv:1908.00821*, 2019.

[19] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “Enet: A deep neural network architecture for real-time semantic segmentation,” *arXiv preprint arXiv:1606.02147*, 2016.

[20] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial as deep: Spatial cnn for traffic scene understanding,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

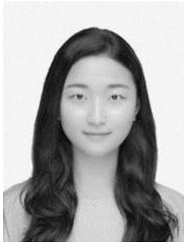
[21] L.-N. Teow and K.-F. Loe, “Robust vision-based features and classification schemes for off-line handwritten digit recognition,” *Pattern Recognition*, vol. 35, no. 11, pp. 2355–2364, 2002.

[22] C. Li, W. Zhang, G. Peng, and S. Liu, “Bearing fault diagnosis using fully-connected winner-take-all autoencoder,” *IEEE Access*, vol. 6, pp. 6103–6115, 2017.

[23] kivinju. tusimple benchmark ground truth. [Online]. Available: https://github.com/TuSimple/tusimple-benchmark/issues/3#issue-372217074

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3078849, IEEE Access



EUNBIN SEO was born in Suwon, South Korea, in 1999. She is currently pursuing the B.S. degree in computer science and technology with Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea. Her research interests include computer vision and automatic control.

SEUNGGI LEE was born in Seoul, South Korea, in January 2000. He is currently pursuing the B.S. degree in electrical engineering from Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea. His research interest include communication system and wireless power transfer.

GWANJUN SHIN was born in Seosan, South Korea in 1999. He is currently undergraduate students at College of Trans-disciplinary studies, Daegu Gyeongbuk Institute of Science and Technology (DGIST). His research interests include visual recognition and domain adaptation in machine learning. He is also the NVIDIA Deep learning institute University Ambassadorship and instructor.

HOYEONG YEO was born in Daegu, South Korea in 1999, He is currently undergraduate students at College of Trans-disciplinary studies, Daegu Gyeongbuk Institute of Science and Technology (DGIST). His research interests include motion control and deep learning.

YONGSEOB LIM received the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2010. He was with the Hyundai Motor Company R&D Advanced Chassis Platform Research Group, South Korea, as a Research Engineer. His works at Hyundai Motor Company focused on advanced vehicle dynamics and control technologies, including active suspension, steering and braking control systems. Moreover, He also with Samsung Techwin Company Mechatronics System Development Group, South Korea as a Principal Research Engineer. His works at Samsung focused on developing stabilization control algorithm for the Remotely Controlled Robotic Arms. He is currently an Associate Professor in the Robotics Engineering Department, Daegu Gyeongbuk Institute of Science and Technology (DGIST), and the Co-Director of the Joint Research Laboratory of Autonomous Systems and Control (ASC) and Vehicle in the Loop Simulation (VILS) Laboratories. His research interests include the modeling, control, and design of mechatronics systems with particular interests in autonomous ground vehicles and flight robotics systems and control.

GYEUNGHO CHOI received the Ph.D. degree in mechanical engineering from the University of Alabama, AL, USA, in 1992, also Honorary PhD Degree in Automotive & Energy Engineering conferred be Her Royal Highness Princess Maha Chakri Sirindhorn from King Mongkut's University of Technology North Bangkok, Thailand in 2018. He served President in Korean Auto-vehicle Safety Association and Editor-in-Chief in Korean Society for Engineering Education. He has taught Thermodynamics, Heat Transfer, and Automotive Engineering in Keimyung University and Daegu Gyeongbuk Institute of Science and Technology as a Professor for 28 years. He is also the Co-Director of the Joint Research Laboratory of Autonomous Systems and Control (ASC) and Vehicle-in-the-Loop Simulation (VILS) Laboratories. His research interests include self-driving vehicle, ADAS system, Vehicle-In-the-Loop system, and design of alternative energy utilization.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/