Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles

Keonyup Chu, *Student Member, IEEE*, Minchae Lee, *Student Member, IEEE*, and Myoungho Sunwoo, *Member, IEEE*

Abstract—**In this paper, a real-time path-planning algorithm that provides an optimal path for off-road autonomous driving with static obstacles avoidance is presented. The proposed planning algorithm computes a path based on a set of predefined waypoints. The predefined waypoints provide the base frame of a curvilinear coordinate system to generate path candidates for autonomous vehicle path planning. Each candidate is converted to a Cartesian coordinate system and evaluated using obstacle data. To select the optimal path, the priority of each path is determined by considering the path safety cost, path smoothness, and path consistency. The proposed path-planning algorithms were applied to the autonomous vehicle A1, which won the 2010 Autonomous Vehicle Competition organized by the Hyundai–Kia Automotive Group in Korea.**

Index Terms—**Autonomous vehicle, nonholonomic constraints, obstacle avoidance, real-time path planning, static obstacles.**

I. Introduction

INTELLIGENT vehicle technologies have the potential for improving driving safety and innovating driving automation for the automobile industry. Current research on driving automation is largely dedicated to the intermediate goal of driving assistance, such as adaptive cruise control and lane-keeping systems. In existing advanced driver-assistance systems, the driver remains solely responsible for the control of the vehicle. However, electronic control systems have become reliable enough to provide greater driving assistance. Autonomous driving capabilities for a limited driving situation can be realized in the near future [1]–[3].

Autonomous driving technology is an emerging area of research that seeks to achieve the ultimate in automobile safety and comfort. This field has received great interest from academia, industry, and the military. Demonstrations on California highways in 1997 proved that autonomous driving is possible if restricted to a dedicated lane on a highway with magnets [2]. In recent years, the Defense Advanced Research Projects Agency (DARPA) Grand Challenge and Urban Challenge have stimulated research interest in the field. Although vehicles that were developed for the DARPA challenges exhibit great autonomous driving performance, significant challenges in terms of technology and price still remain for the commercialization of autonomous vehicles [2], [4]–[6].

The development of autonomous vehicles requires state-of-the-art technologies in perception, planning, and control. In particular, planning allows an autonomous vehicle to determine the behavior of the vehicle by itself. Planning algorithms for autonomous vehicles can be divided into global and local stages. In the global planning stage, the global route and the vehicle states are determined through information from a digital map and a localization system. The local path can then be generated in the local planning stage based on the global route and surrounding information of the vehicle obtained from sensors such as a camera and radar [2], [4]–[7]. In this paper, we focus on local path planning, which provides obstacle avoidance capabilities to an autonomous vehicle following a predefined global route.

Many studies have been conducted on path generation. Although path generation was initially studied for robotic applications, it has more recently been studied for automotive applications. The objective of path planning is to search for the optimal motion of a vehicle subject to obstacle fields and vehicle motion constraints. Research on path planning can primarily be separated into potential-field approaches, grid-based approaches, sample-based approaches, and discrete optimization approaches.

Potential-field algorithms virtually assign repulsive forces to obstacles and attractive forces to the goal position. Virtual forces are used to form the gradient of a potential field. A path can then be found along the steepest gradient of the potential field. However, the trajectory can be trapped in the local minima of the potential field in several scenarios, and finding a path is not guaranteed [8], [9].

In grid-based approaches, the environment is mapped to a set of cells, where each cell represents the presence of an obstacle at that position in the environment. Optimal search methods, such as the A* and D* algorithms, are frequently used to find a globally optimal path that connects each cell from an initial position to a goal while avoiding obstacles in the environment.

Manuscript received April 23, 2011; revised October 7, 2011 and February 21, 2012; accepted April 20, 2012. Date of publication May 22, 2012; date of current version November 27, 2012. This work was supported in part by the Ministry of Education, Science and Technology through the BK21 Program under Grant 201000000000173, the Ministry of Knowledge Economy (MKE) and Korea Institute for Advancement of Technology through the Workforce Development Program in Strategic Technology, Energy Resource R&D Program under Grant 2006ETR11P091C, the National Research Foundation of Korea, funded by the Korean government through the Ministry of Education Science and Technology, under Grant 2011-0017495, and the Industrial Strategy Technology Development Program of MKE under Grant 10039673. The Associate Editor for this paper was B. De Schutter.

K. Chu and M. Lee are with the Automotive Control and Electronics Laboratory, Department of Automotive Engineering, Hanyang University, Seoul 133-791, Korea (e-mail: acehev@hanyang.ac.kr, eagerid@gmail.com).

M. Sunwoo is with the Department of Automotive Engineering, Hanyang University, Seoul 133-791, Korea (e-mail:msunwoo@hanyang.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2012.2198214

The resolution of the grid cells involves a tradeoff between the optimality of a path and the practical efficiency of computation and memory usage to find a path. In the traditional grid-based approach, it is difficult to treat the nonholonomic constraint of the vehicular system. Recently, several variants of the grid-based algorithm that treat the nonholonomic constraints have been studied by adapting applications such as executing U-turns on a blocked road, parking, and navigating a road in unstructured environments [10]–[12]. However, the incremental nature of search algorithms induces the exponential growth of the memory usage with respect to the complexity of the problem. Therefore, although grid-based approaches are suitable for applying path planning to low-speed applications, they are not are suitable for high-speed driving.

Sampling-based planning algorithms are appropriate for planning problems in high-dimensional space. In sampling-based approaches, the algorithm constructs a collision-free path from the initial configuration to the destination by sampling the configuration that describes the position and orientations of the vehicle. Among the various sampling-based approaches, rapidly exploring random tree (RRT) and variants of RRT are widely used in nonholonomic motion planning, such as in automotive applications. RRTs are incrementally constructed in a manner that quickly reduces the expected distance from a randomly chosen point to the tree. RRT-based planning algorithms for real-time implementations require efficient guiding heuristics for the sampling configuration [9], [13]–[16].

Recently, several groups have achieved great success in autonomous driving based on path planning through discrete optimization in DARPA challenges. In this scheme, a finite set of paths is computed by the numerical integration of the differential equations that describe the vehicle motion. From this set, an optimal path that minimizes the cost is selected. Applying a finite set of paths reduces the solution space and allows for real-time implementation through fast computation [6], [17]–[19]. In particular, an efficient path-planning algorithm was implemented with an autonomous vehicle Stanley (Stanford) during the DARPA Grand Challenge. Stanley's path planner defines the lateral offset as the perpendicular distance to a fixed base trajectory. This condition enables the autonomous vehicle to drive the road parallel to the base trajectory. To select the optimal path, the cost function penalizes running over obstacles and distance from the current road center.

In this paper, we develop a path-planning algorithm that selects a local path from a finite set of paths. Our approach is conceptually similar to the path planning used by Stanley (Stanford) in the 2005 DARPA Grand Challenge in terms of using a base frame and generating path candidates with endpoints at fixed offsets from the base frame. The global route forms a base frame of a curvilinear coordinate system that is the design space for the path. To generate path candidates, the directional information of the global route is blended to the maneuvering of the vehicle by adjusting the lateral offset to the base frame. Our approach for path planning is distinguishable from Stanley's approach in terms of path selection criteria.

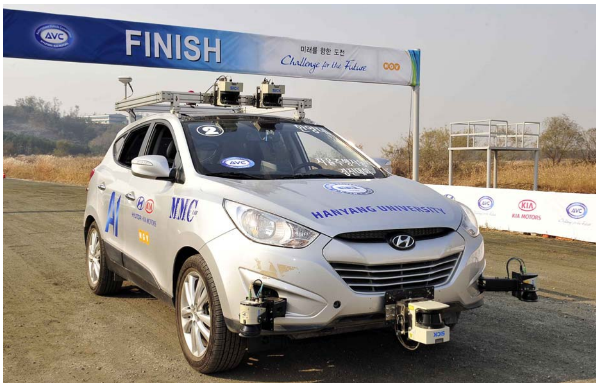


Fig. 1. A1: The autonomous vehicle of Hanyang University, Seoul, Korea, at the AVC.

Stanley's approach directly uses an estimated road center based on Kalman filters for the path selection, whereas the proposed path-planning algorithm utilizes the safety cost for each path candidate. Because the road center estimation may be ambiguous and heuristic for evaluating the safety of the path in some scenarios, the proposed algorithms do not directly distinguish the road center. Instead, the safety of each path is quantitatively evaluated in blurring the binary data for a collision. Therefore, it is not necessary to accurately estimate the road center in the proposed approach. In addition, the smoothness and the consistency of the path are also applied to the optimization in this algorithm (the cost functions adapted to Stanley penalize running over obstacles and distance from the road center.). This approach enables the autonomous vehicle to more stably drive itself. Through this paper, the effects of each cost function are presented and analyzed for the intuitive design of the weighted cost function.

In this paper, we focus on the description of the local path planning for autonomous vehicles in a practical way, with experiments rather than theoretical improvement of the path-planning algorithm. Although almost all of the concepts applied in this paper, such as the utilization of curvilinear coordinates and cost functions for path selection, are individually not novel, the integration of these concepts yields successful results in the experiments. The proposed path-planning algorithms were applied to the autonomous vehicle A1, which won the 2010 Autonomous Vehicle Competition (AVC) organized by the Hyundai–Kia Automotive Group in Korea. The vehicle, which was developed by the Automotive Control and Electronics Laboratory (ACE Lab) and the Machine Monitoring and Control Laboratory (MMC Lab), Hanyang University, Seoul, Korea, is shown in Fig. 1.

The remainder of this paper is organized as follows. The system architecture of the autonomous vehicle A1 and the structure of the path-planning algorithm are briefly introduced in Section II. A description of the base frame, which provides the horizontal axis of the curvilinear coordinate system for the path planning, is given in Section III. In Section IV, the design of the path candidates is described in the curvilinear coordinate system for the base frame. The selection criteria for the smoothness and safety of the path are explained in Section V, and the determination of the target speed is briefly shown. The experimental results that were obtained in an AVC are presented in Section VII. This paper is ultimately concluded in Section VIII.

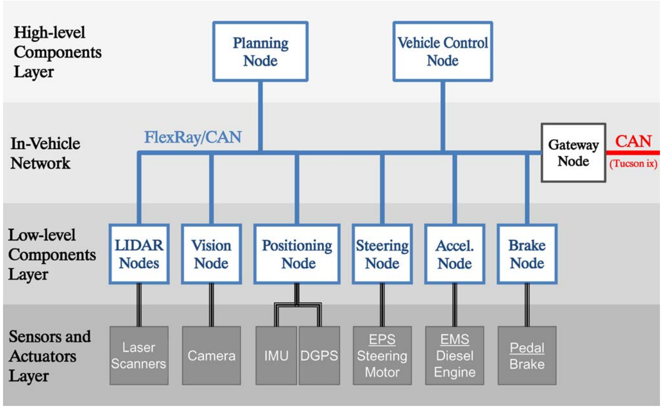


Fig. 2. Network-based system architecture of A1.

II. System Overview

A. System Architecture

The autonomous driving system of A1 has a layered architecture based on the in-vehicle network (IVN), as shown in Fig. 2: a high-level components layer; a low-level components layer; a sensor and actuator layer; and an IVN. This network-based layered architecture provides the distributed system architecture of an autonomous driving system. This distributed computing system of A1 encapsulates the logical level function and the associated hardware into a subsystem, i.e., a node. Each subsystem of A1, called a node, was specified and tested by considering the functional and temporal behavior.

The role of the high-level components layer is the integration of all of the information from the perception nodes and the decision making. This layer consists of the following two nodes: 1) the planning node and 2) the vehicle control node. The planning node receives the vehicle states and environment information from the low-level components layer through the IVN, and it generates a drivable path and a target vehicle speed. It also determines the driving mode and provides the user interface function. In the vehicle control node, the practical control commands are computed from the path given by the planning node. The vehicle control node decides the desired steering angle, acceleration, and brake. These decisions are transmitted to the steering, acceleration, and brake nodes in the low-level components layer through the IVN.

The low-level components are composed of several nodes to process the raw data from the sensors and also to control the actuators. The low-level components layer has the following three kinds of perception nodes: 1) the light detection and ranging (LIDAR) nodes; 2) the vision node; and 3) the positioning node. The LIDAR node interacts with the laser radar sensor that provides the range data of the obstacles. Using the range data, the LIDAR node filters out the cluttered data and abstracts the obstacle information. The vision node is connected to the single camera and detects lane and crosswalk in the paved road from the image data. For the control of the vehicle motion, the steering, acceleration, and brake nodes are connected to individual actuators. The sensor and actuator components layer are composed of laser radars, a camera, an inertia measurement unit (IMU), a Differential Global Positioning System (DGPS), an electric power steering motor, a diesel engine, and a brake.

In a distributed real-time system, system integration is accomplished by communication between the different nodes. For this reason, the IVN has a significant role in the integration of the distributed autonomous driving system. The IVN consists of a controller area network (CAN), FlexRay, Ethernet, Gateway, and a conventional IVN. In our application, the CAN is used as the primary network protocol for the communication between multiple nodes of the autonomous driving systems. The purpose of the gateway in our application is to transmit the vehicle sensor data, such as steering angle and wheel speed, from the conventional IVN of a Hyundai Tucson ix to the nodes in the autonomous driving system.

To construct the autonomous driving system in Fig. 2, the dedicated embedded systems are developed for most electronic systems for the perception and control systems of the A1. For the development of the embedded system hardware, two kinds of microcontrollers, the HCS12XF and MPC5567 from Freescale Semiconductor, have been used. HCS12XF is a 16-b microcontroller that is suitable for the automotive chassis and body application. In our application, HCS12XF is mainly applied in low-level applications, such as the steering angle control, acceleration, data parsing from the DGPS receivers, and the emergency stop interface. MPC5567 is a 32-b microcontroller that has mostly been used in automotive powertrain applications that require high computation power. In our application, MPC5567 is used in the perception system such as the positioning system and the obstacle detection systems. For computer vision and planning, an industrial personal computer with an Intel Core 2 Duo central processing unit is used, because the processing of the computer vision algorithm and path planning require extremely high performance in terms of the memory and computation time compared with the other applications.

For a measure of the global position of the vehicle, two DGPSs were installed on the roof of the A1, as shown in Fig. 3. Instead of directly using the measurement data from the DGPSs, a multiple-model-based positioning method was employed to estimate the vehicle position and orientation by integrating the DGPS data. To consider the Global Positioning System (GPS) blockage and poor GPS information, the positioning system of A1 checks the number of satellites used for position estimation using the GPS and the horizontal dilution of precision (HDOP). The estimation algorithm does not update the GPS measurements when the number of satellites in use is less than or equal to four or when the HDOP is greater than five. In this case, the vehicle model plays an important role in estimating the vehicle position using the wheel speed sensor, steering angle sensor, and IMU (see Table I). The positioning system that was applied to A1 provided continuous and reliable position information, even in a harsh GPS signal environment [20], [21].

In autonomous driving, it is essential to detect obstacles so that they can be avoided. For obstacle detection, two laser-radars that look down the road ahead were installed on the roof of A1, and three laser-radars that scan parallel to the vehicle were mounted on the bumper, as shown in Fig. 3. The

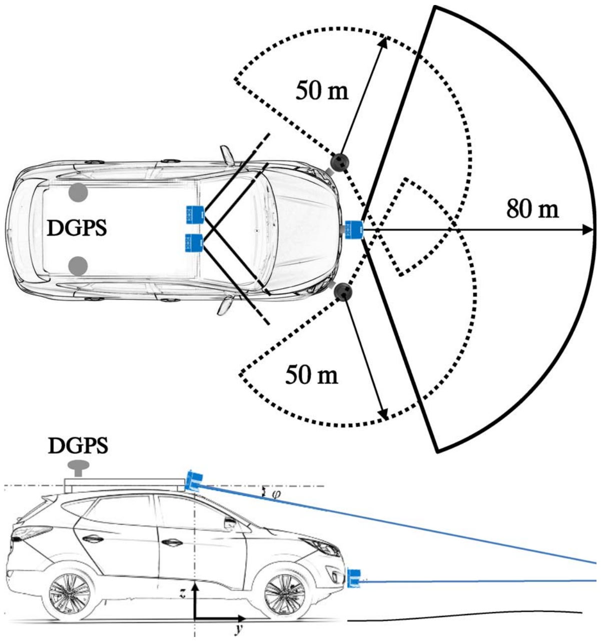


Fig. 3. Sensor installation of A1.

TABLE I
SENSORS INCORPORATED INTO A1

Sensor	Characteristics and configurations
SICK LMS 291	100 deg FOV with 0.25 deg angular resolution 80 m maximum range
SICK LMS 151	270 deg FOV with 0.5 deg angular resolution 50 m maximum range
HUACE DGPS B20	0.75 m RMS horizontal error 12x L1 GPS channels, 10 Hz output rate
Xsense MTi-G	GPS aided AHRS 3D acceleration with ±50 m/s^2 full scale 3D rate of turn with ±300 deg/s full scale
Imaging source DFK 21BUCO3	USB CMOS Color Camera 1/3 Micron CMOS, progressive scan Up to 744x480 pixel

three laser-radars that were mounted on the front bumper have different ranges: the maximum range of the center laser radar is 80 m, whereas the maximum range of the side laser radars is 50 m. Data from each laser-radar were processed to remove erroneous information such as ground clutter caused by the pitch motion of the vehicle [22]. The processed obstacle data were used to construct the cell-based local map, which is utilized by the path planner of the A1 to evaluate the path candidates. For the evaluation of the path candidates, the maximum distance of path was limited to 50 m, which relies on the detection distance of the laser scanner of the A1.

B. Structure of the Path-Planning Algorithm

The goal of the path-planning algorithm of A1 is to generate a smooth path from the initial vehicle configuration toward the direction of the global route. An illustration of the concept of path planning is shown in Fig. 4. The path-planning algorithm generates path candidates with different offsets from the base frame. The algorithm is decomposed into the following four components.

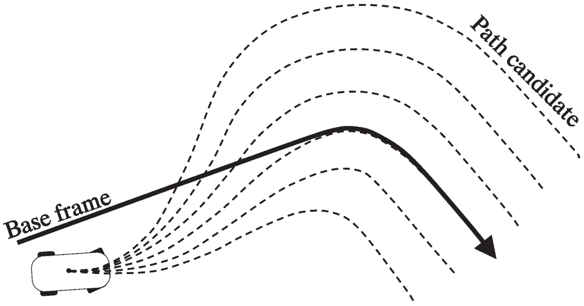


Fig. 4. Base frame and candidate paths.

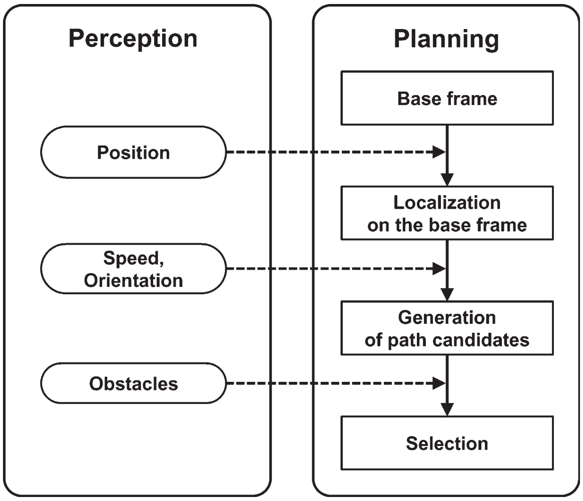


Fig. 5. Path-planning strategy.

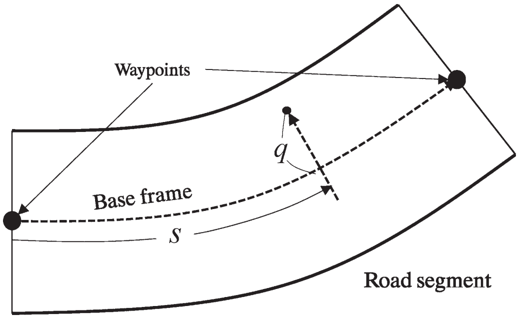


Fig. 6. Road segment and curvilinear coordinate system.

1) construction of the base frame;
2) localization on the base frame;
3) generation of path candidates;
4) selection.

These components are shown in Fig. 5. The base frame is constructed by a spline, which is based on a set of waypoints that contains the center position of the road, as shown in Fig. 6. To utilize the road shape from the base frame, the position of the vehicle is determined on the base frame during the localization step. The path candidates are then generated by the forward integration of the model using the base frame, current vehicle speed, and orientation. From this set of paths, one path is selected, which minimizes the cost function of the

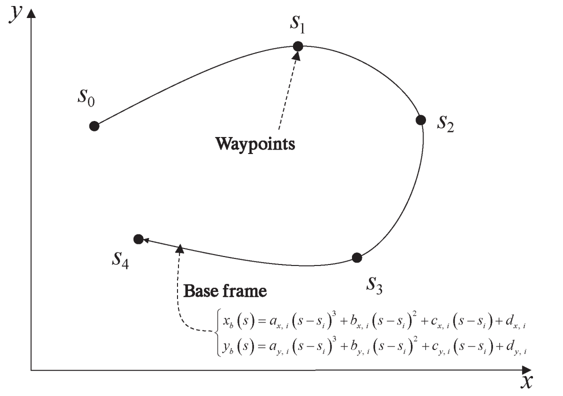

y
s_1
s_0
Waypoints
s_2
s_4
Base frame
s_3
$x_b(s)=a_{x,i}(s-s_i)^3+b_{x,i}(s-s_i)^2+c_{x,i}(s-s_i)+d_{x,i}$
$y_b(s)=a_{y,i}(s-s_i)^3+b_{y,i}(s-s_i)^2+c_{y,i}(s-s_i)+d_{y,i}$
x

Fig. 7. Waypoints and base frame.

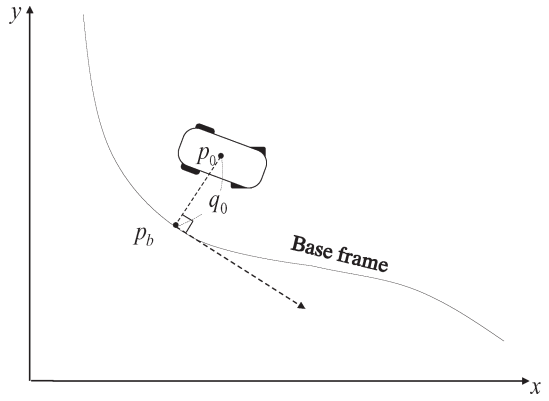


Fig. 8. Localization on the base frame.

path candidates. In the selection stage, the risk of collision and the smoothness of the path are considered through the use of obstacle data. The path-planning algorithm is executed at 20 Hz, and it generates a new path from the current pose of A1 at every time step.

III. Base Frame for the Curvilinear Coordinate System

A. *Approximation of Arc-Length Parameterization for the Base Frame*

Parametric curves are frequently employed to model the geometry of a road and to define motion paths. Among all of the different approaches to represent a road, a parametric cubic spline is utilized here to define the parametric curve of the base frame that is constructed based on waypoints, as shown in Fig. 7. In this paper, it is assumed that precise waypoints are given to omit global path planning. The position on the base frame is directly computed from the parametric cubic spline curve that corresponds to the parameter. However, this parameter has no physical meaning, and it is difficult to directly apply these parametric curves as the base frame of the motion planning for the autonomous vehicle. If the base frame is an arc-length parameterized curve, the formulations of the maneuvering for obstacle avoidance and vehicle path tracking are simplified. For this reason, it is essential to relate parameters to the arc length of the curve to achieve real-time performance for path generation. For the given waypoints, a curve that represents the base frame is parameterized using an arbitrary parameter to form a spline curve. To parameterize the spline curve as the arc length, the arc lengths of all of the segments in the spline curve are numerically computed using an adaptive Gaussian quadrature. The spline curve for the base frame is then again parameterized by the approximated arc length of the curve [23], [24]. The 2-D parametric cubic spline that was used for the base frame, with s being the arc length, is expressed as

$$\begin{cases} x_b(s)=a_{x,i}(s-s_i)^3+b_{x,i}(s-s_i)^2+c_{x,i}(s-s_i)+d_{x,i} \\ y_b(s)=a_{y,i}(s-s_i)^3+b_{y,i}(s-s_i)^2+c_{y,i}(s-s_i)+d_{y,i}. \end{cases} \tag{1}$$

If the parameter s is given, the corresponding position $[x_b, y_b]^{\mathrm{T}}$ on the base frame is directly computed from the parametric cubic spline curve in (1). The tangent and curvature of the base frame can easily be calculated by the first and second derivatives of the arc-length parameterized cubic spline curve as [25]

$$\frac{dx_b}{ds} = x_b' = \cos\theta_b \quad \frac{dy_b}{ds} = y_b' = \sin\theta_b \tag{2}$$

$$\frac{d^2x_b}{ds^2} = x_b'' \quad \frac{d^2y_b}{ds^2} = y_b'' \tag{3}$$

$$\kappa_b = \frac{x_b'y_b'' - x_b''y_b'}{(x_b' + y_b')^{3/2}}. \tag{4}$$

The parametric cubic spline provides the continuity of the second derivative, which involves the continuity of curvature in all sections. For this reason, the parametric cubic spline curve is adopted as a representation of the base frame in this paper.

B. *Localization on the Base Frame*

To utilize the directional information of the base frame, mapping from the Cartesian to the curvilinear coordinates is required. The key component in this mapping is the computation of the closest point on the base frame to a point that is expressed in the Cartesian coordinates. The closest point of the vehicle position on the base frame is numerically computed through a two-stage technique that combines quadratic minimization and Newton's method. Such a scheme minimizes the distance q_0. In this process, the arc length of the closest point on the base frame is found to minimize the distance between the current position p_0 and the position p_b on the base frame, as shown in Fig. 8 [26].

IV. Generation of Path Candidates

A. *Transformation Between Curvilinear Coordinates and Cartesian Coordinates*

The path must guide the vehicle in following the global route and avoiding obstacles. These tasks are accomplished by setting up a curvilinear coordinate system of the base frame. Smooth maneuvering for obstacle avoidance is designed in the curvilinear coordinate system of the base frame rather than in

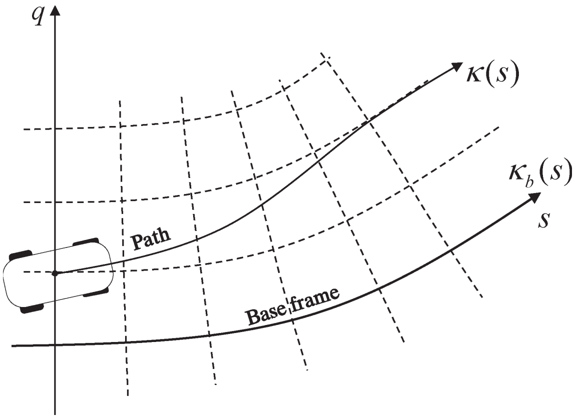


Fig. 9. Relation of the path and the base frame.

the Cartesian coordinate system. The geometric relationship between the path and the curvilinear coordinate system of the base frame in the Cartesian coordinate system is shown in Fig. 9. In this figure, the arc length of the base frame s, which is the traveling distance on the global route, becomes the horizontal axis of the curvilinear coordinate system, whereas the lateral offset q becomes the vertical axis. For the transformation between the Cartesian coordinate system and the curvilinear coordinate system, the curvature of the path κ is derived as [18], [27]

$$\kappa = \frac{S}{Q}\left(\kappa_b + \frac{(1-q\kappa_b)(d^2q/ds^2)+\kappa_b(dq/ds)^2}{Q^2}\right) \quad (5)$$

with

$$S = \operatorname{sgn}(1-q\kappa_b), \quad Q = \sqrt{\left(\frac{dq}{ds}\right)^2 + (1-q\kappa_b)^2}. \quad (6)$$

If the lateral offset q is the same as the radius of curvature of the base frame $1/\kappa_b$, this path passes through the center of the curvature of the base frame. If the lateral offset q is larger than the radius of curvature of the base frame, $1/\kappa_b$ the curvature and the direction of the generated path will be opposite in sign to the base frame. In this case, the generated path is discarded from the path candidates, because it violates the nonholonomic constraint of vehicle motion. Therefore, feasible path generation is guaranteed only when the lateral offset of the path is less than the radius of curvature of the base frame.

The curvature of the path is directly related to the vehicle motion, which can be described through the use of various vehicle models. The model of a real vehicle would require more physical effects of mass and inertia, but these terms do not affect the geometric shape of the path. Therefore, a simplified kinematic model is given by [27]

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = v\kappa \quad (7)$$

where $[x\ y\ \theta]^{\mathrm{T}}$ is the position and orientation of the vehicle, v is the vehicle speed, and κ is the curvature of the path. The curvature of the vehicle is limited by a steering angle constraint. This constraint for the curvature is given by

$$|\kappa| \le \kappa_{\max}. \quad (8)$$

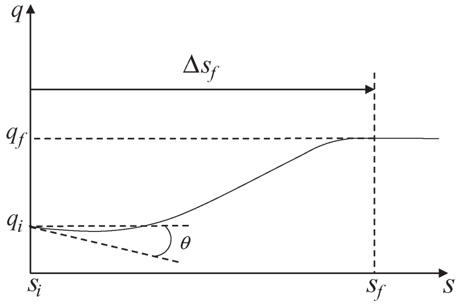


Fig. 10. Maneuver in the curvilinear coordinate system.

If this constraint for the curvature of a path candidate is violated, the corresponding path is thrown out. According to (7), the motion of the vehicle has only two degrees of freedom as follows: 1) speed and 2) curvature. Because we focus only on the geometric formation of the path at this stage, the vehicle speed v is eliminated from the vehicle model. To represent vehicle motion in terms of the traveling distance rather than time, the vehicle speed v is given by [27]

$$v = SQ\frac{ds}{dt}. \quad (9)$$

By substituting the vehicle speed into the simplified kinematic model (7), the differential equation of motion can be represented with respect to the arc length of the base frame as

$$\frac{dx}{ds} = Q\cos\theta, \quad \frac{dy}{ds} = Q\sin\theta, \quad \frac{d\theta}{ds} = Q\kappa. \quad (10)$$

This approach allows the path to be generated independent of the vehicle speed along the base frame. Therefore, a path can be determined by the forward integration of this model with respect to the arc length s on the base frame.

B. Maneuvering in the Curvilinear Coordinate System

To generate a path, the curvature of the path is determined by the lateral offset q of the path based on the curvature of the base frame. The derivative and the second derivative of the lateral offset are also required to calculate the curvature of the path. Therefore, a lateral offset function is designed to provide a smooth change in the lateral offset. The maneuver for a change in the lateral offset in the curvilinear coordinate system is shown in Fig. 10. The lateral offset q is described by a sequence of a cubic polynomial and a constant, i.e.,

$$q(s) = \begin{cases} a\Delta s^3 + b\Delta s^2 + c\Delta s + q_i, & s_i \le s < s_f \\ q_f, & s_f \le s \end{cases}$$

$$\frac{dq}{ds}(s) = \begin{cases} 3a\Delta s^2 + 2b\Delta s + c, & s_i \le s < s_f \\ 0, & s_f \le s \end{cases}$$

$$\frac{d^2q}{ds^2}(s) = \begin{cases} 6a\Delta s + 2b, & s_i \le s < s_f \\ 0, & s_f \le s \end{cases} \quad (11)$$

where $\Delta s = s - s_i$.

In Fig. 10, the arc length s_i and the lateral offset q_i at the current position are given in the localization stage. The angle difference θ is defined by the difference between the vehicle heading angle and the tangent angle of the base frame at the current position. The arc length s_f and the lateral offset q_f at the end position of the change in the lateral offset are design parameters for the generation of various paths. Based on these relationships, the boundary conditions for the cubic polynomial are determined as follows [28], [29]:

$$\begin{aligned} q(s_i) &= q_i, \quad \frac{dq}{ds}(s_i) = \tan\theta \\ q(s_f) &= q_f, \quad \frac{dq}{ds}(s_f) = 0. \end{aligned} \tag{12}$$

Based on these boundary conditions, the coefficients of the cubic polynomials are determined.

C. Generation of Candidate Paths

The path-planning algorithm generates a finite number of path candidates, each of which has a distinct lateral offset q_f. All of the candidate paths cover the road with a change in the lateral offset. The lateral offset change in the path enables an autonomous vehicle to avoid obstacles.

The maneuvering for a lateral offset change is determined by specifying an arc-length distance of the base frame Δs_f to complete lateral offset maneuvering in Fig. 10. The distance of the lateral offset change Δs_f and the vehicle speed v dominantly affect the rate of lateral offset change. A short Δs_f induces a geometrically steep change in the lateral offset for the path generation, whereas a long Δs_f yields a smooth change in the lateral offset. For the same distance of lateral offset change, a fast vehicle speed gives a fast rate of lateral offset change, which causes an increase in the lateral acceleration. An increase in the lateral acceleration may degrade the comfort and stability of driving. Therefore, a fast vehicle speed requires a long distance for a change in the lateral offset. The distance of the lateral offset change Δs_f is determined as a function of the speed as follows:

$$\Delta s_f = k_v v + \Delta s_{\min} \tag{13}$$

where k_v is the proportional gain for the current vehicle speed, and $\Delta s_{\min}$ is the minimum distance for the lateral offset change. The k_v and $\Delta s_{\min}$ terms are heuristic design parameters that are carefully chosen according to the autonomous driving style and the kinematic limitation of the steering. For example, sporty driving requires quick change maneuvering in the lateral offset, which is achieved with a small value of k_v.

Examples of the path candidates are shown in Fig. 12. For fast vehicle speed such as 60 km/h, as shown in Fig. 12(a), the lateral offset change in the path is slow. In contrast, for slow speed such as 40 km/h, as shown in Fig. 12(b), the lateral offset change is steeper. Path candidates in the case of a different orientation between the vehicle and the base frame are shown in Fig. 12(c). The orientation difference does not affect the parallel direction of the path to the base frame without violation of the nonholonomic constraint of the vehicle motion. For a curved road, the path-planning algorithm generates path candidates parallel to the base frame without additional treatment of the curve, as shown in Fig. 12(d). These properties of the path candidates enable the vehicle motion to be natural during autonomous driving.

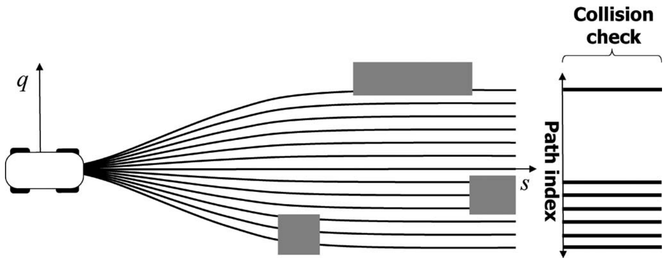


Fig. 11. Collision check for the generated paths.

In terms of the implementation of the path generation scheme, the range and the resolution of the lateral offset are important design factors for the real-time performance of the path-planning algorithm. The range of the lateral offset was determined to be 20 m from side to side at the lateral offset of the current position when considering the road width of the 2010 AVC track. The resolution of the lateral offset involves a tradeoff between the computation time and the resolution of the solution space for the path generation. Although a high resolution for the lateral offset improves the optimality of the solution, it requires more memory usage and computation time. The resolution of the lateral offset was experimentally determined to be 0.1 m to satisfy the period of the path generation with sufficient resolution.

V. Online Path Selection

For the selection of a path, a search algorithm finds a path that minimizes a linear combination of the cost functions with weighting factors. The cost functions evaluate the safety, smoothness, and consistency of the path as follows:

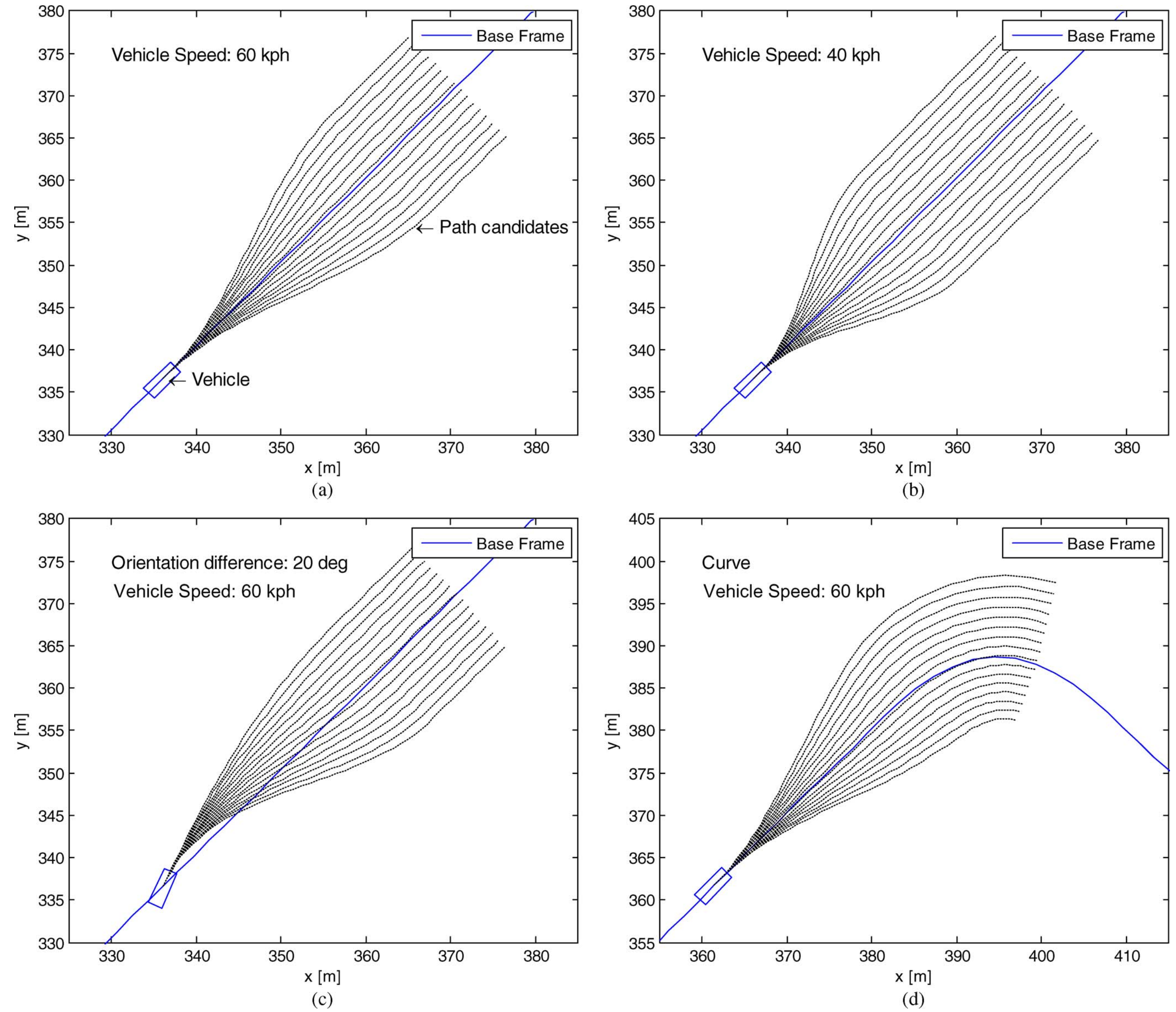
$$J[i] = w_S C_S[i] + w_\kappa C_\kappa[i] + w_C C_C[i] \tag{14}$$

where i is the index of a path, C_S is the cost for the safety of a path, C_κ is the cost for the smoothness of a path, and C_C is the cost for path consistency. Each cost has a weighting factor that determines the characteristics of the path.

A. Cost for the Safety of Paths

To prevent collision, a collision check for each path candidate is required. A path that runs over obstacles must be removed from the candidates. One example of the collision check on several path candidates generated in the previous stage is shown in Fig. 11. On the right side of Fig. 11, the result of the collision check on the path candidates is represented. If the shortest path criterion is adopted to select an optimal path, the central path may be chosen in Fig. 11. This path is adjacent to an obstacle and is not desirable for safe autonomous driving. Adjacent driving to an obstacle may increase the probability for a collision caused by a control error and occluded obstacles.

It is difficult to select a safer path from the path candidates by solely checking the collision for each maneuver, because the collision check provides only binary information. Therefore,



(a)

(b)

(c)

(d)

Fig. 12. Path candidates.

a fast risk assessment for collision is required for every path candidate. The distance between a path and an obstacle provides good information for assessing the safety of the path. However, it is impractical to calculate the distance between a path and obstacles when there are a large number of obstacles in the obstacle map. Instead, our approach is to explicitly compute the distances of the obstacles: The risk for each path is quantitatively evaluated in blurring the binary data for a collision. The level of risk is ambiguous and subjective for path evaluation. To infer the risk of a path, the following subjective properties are used: The collision affects the risk of a neighbor of the path, and the collision has an insignificant effect on the risk of a distant path. The risk of a path is defined by discrete Gaussian convolution with collisions as follows:

$$C_S[i] = \sum_{k=0}^{N} c[k]g[i-k] \tag{15}$$

where

$$g[i] = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\Delta q \cdot i)^2}{2\sigma^2}\right). \tag{16}$$

In addition, $c[i]$ is the collision detection of a path with index i, Δq is the resolution of the lateral offset for a path, and σ is the standard deviation of the risk of collision. The basic concept of Gaussian convolution for the risk of a path is shown in Fig. 13. In the case of the convolution for an out-of-path index, the detection of a collision is always regarded as true. The standard deviation of the Gaussian kernel $g[i]$ is a design factor in determining the effective range of the collision detection for each path. The result of the convolution that gives the distributions of the risk is presented in Fig. 14. The collision detection of each path is shown in Fig. 14(a), whereas the collision risks of the paths are displayed in Fig. 14(b). The horizontal axis of these graphs presents the index of the path

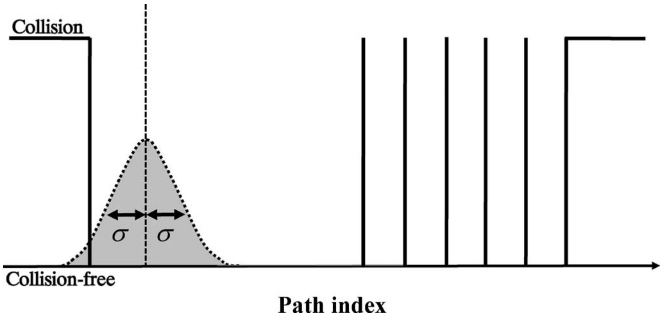


Fig. 13. Gaussian convolution for the risk of the path.

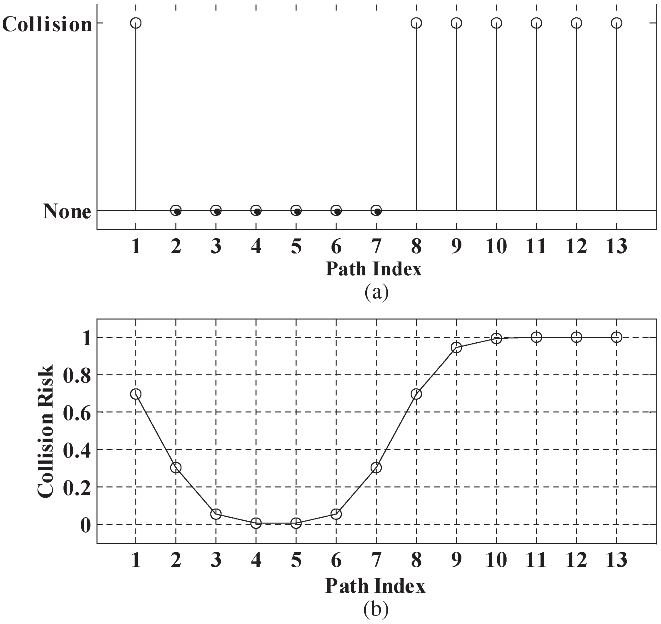


Fig. 14. Collision risk.

that is from the lateral offset of the path from the base frame. Considering the case of Fig. 11, index 7 in Fig. 14 is the index of the center path with the shortest distance to travel on the global route. However, the risk of the path in Fig. 14(b) shows that the center path is risky compared to the other paths with three to six indices, because it is too close to obstacles. The distribution of risk distinguishes the degree of safety of each path and is used as a cost for evaluating the safety of the path. Although interacting with the dynamic environment and the rules of real road are important topics that require further research, we concentrate on the autonomous driving with static obstacles in this paper. However, our approach in this paper can be extended to driving in a dynamic environment using the duality of the arc length and vehicle speed with dynamic collision checking [19], [30]–[33].

B. Cost for the Smoothness of Paths

The smoothness of a path is important for autonomous driving, because it affects the driving quality.

In particular, an unsmooth path may cause slippage of the wheels, which, in turn, degrades the stability of the vehicle motion. Although the notion of smoothness is an ambiguous criterion with which to evaluate a path, the smoothness of a path is related to the curvature profile, which is directly linked to the lateral acceleration of the vehicle. Assuming a constant speed of a vehicle along a path, the instantaneous centripetal acceleration of the vehicle is proportional to the curvature of the path. Therefore, integration of the squared curvature is equivalent to integration of the squared acceleration.

In this paper, the integration of the squared curvature along the length of a path is chosen as the smoothness criterion of a path as follows [34]–[37]:

$$C_\kappa[i] = \int \kappa_i^2(s)\, ds_p = \int \kappa_i^2(s) Q(s)\, ds \tag{17}$$

where s_p is the length along the path, s is the arc length of the base frame, and κ_i is the curvature of a path with index i.

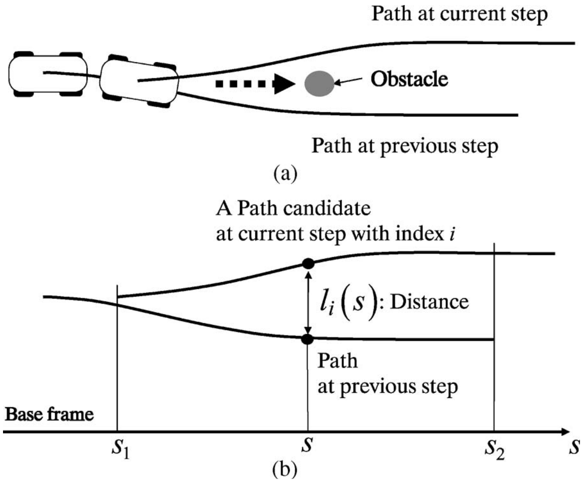


Fig. 15. Difference between a path at the current step and a path at the previous step.

C. Cost for Consistency of the Path

The safety and smoothness costs of the paths are solely calculated with the environmental information that is collected at the current step time. For this reason, the safety and smoothness costs do not prevent the generation of a path that is significantly different from a path at a previous step. An abrupt change in a path requires excessive energy and control effort to follow the path. In particular, if the response of the actuators for the control of the vehicle motion is undesirably slow, an abrupt change in a path may cause a degradation of the driving stability or a collision with obstacles.

One example of a sudden path change that causes a collision with an obstacle is shown in Fig. 15(a). A current path is generated at the left side of an obstacle, whereas a previous path is generated at the right side of an obstacle in Fig. 15(a). The actual vehicle trajectory, which is the result of the tracking paths, may be shown between the previous and the current paths, as shown by the arrow in Fig. 15(a).

In the aforementioned case, it is easier to simply continue in the same direction if an autonomous vehicle fails to avoid obstacles or has an unstable motion with a sudden change in direction. To prevent a significant difference in the path between the path in the current step and the path in the previous step, the consistency of the path must be considered. The concept of path consistency means that current states, actions, or decisions depend on previous states, actions, or decisions. The selection of a current path from the candidates should be performed by considering the past path. To consider the influence of a past path for the path selection, a measure of similarity between the past path and current path candidates is required. In this paper, the integral of the distance between points on the current path and on the previous path along the overlapped region is adopted as a measure of the similarity, as shown in Fig. 15(b). The parameter s_1 is the arc length of the base frame at the starting point of the overlap, and s_2 is the arc length of the base frame at the end of the overlap. The distance of the overlapped region changes according to the vehicle speed. For example, the distance of the overlapped region for a fast vehicle speed becomes shorter. To eliminate the effect of the vehicle speed for the measure of the consistency of a path, the measure of similarity of the path is divided by the arc-length difference between the start and the end of the overlap. The cost for the consistency of a path is defined as follows:

$$C_C[i] = \frac{1}{s_2 - s_1} \int_{s_1}^{s_2} l_i \, ds \tag{18}$$

where l_i is the Euclidean distance between a point in a current path candidate of index i and a point in the path of the previous step with the same arc length of the base frame.

D. Selection of a Path

The selection stage for the path planning plays a significant role in the generation of a safe and reliable path. Furthermore, it is also critical for path selection to handle exceptional situations in an uncertain environment. The entire procedure of the path-planning algorithm, including the path selection, is shown in Algorithm 1. The path-planning algorithm updates the vehicle states and environmental data, such as the current vehicle position, orientation, speed, and obstacles (line 1). The current arc length in the base frame is numerically computed using the directional information of the base frame (line 2). In line 4, the parameters are initialized for a maneuver in the curvilinear coordinate system. Using the initialized maneuvering parameters, a path candidate is calculated by the forward integration of the simplified vehicle model (line 5).

Algorithm 1: Path planning.

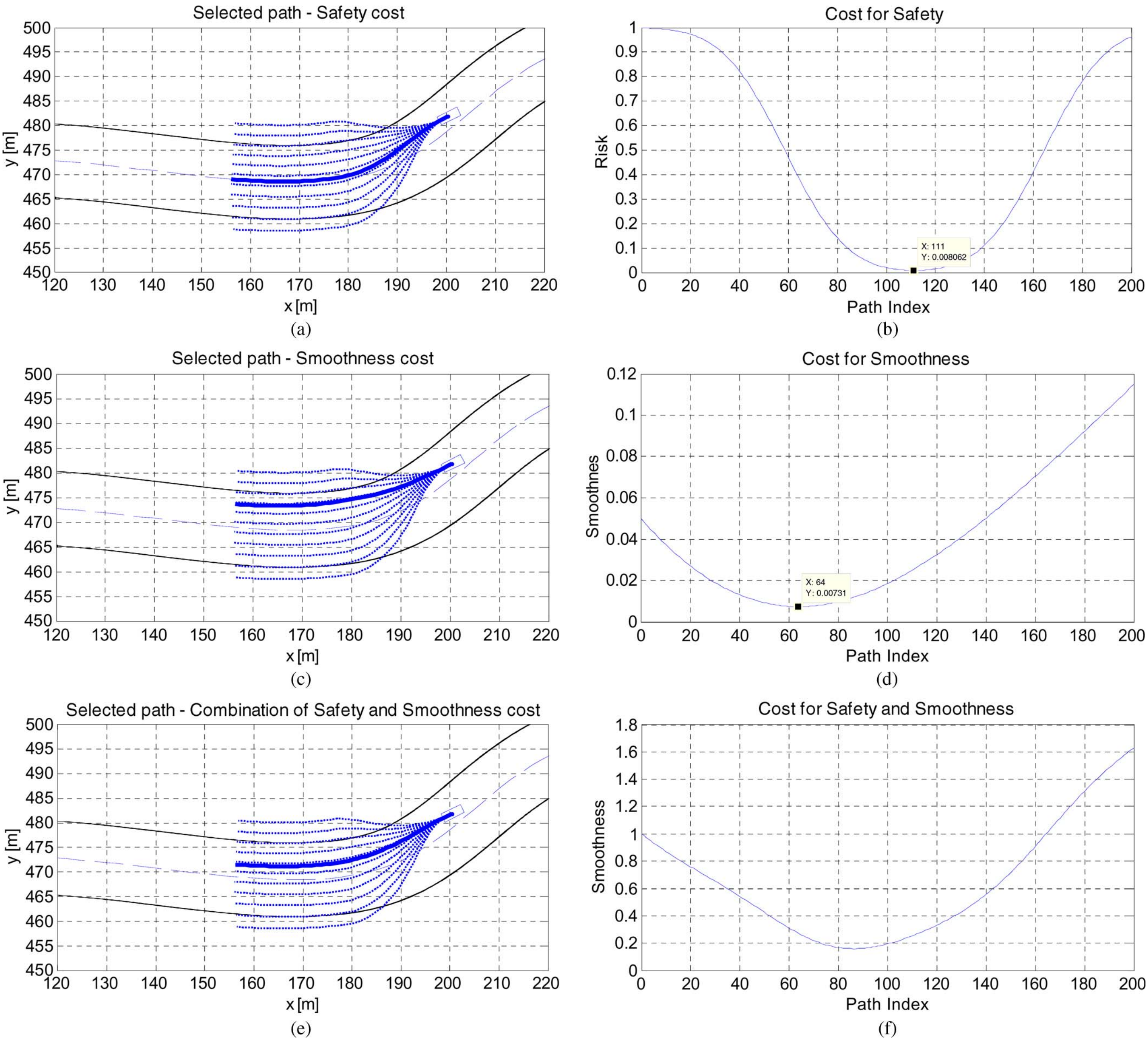
```
1: Update the current vehicle states and environmental data.
2: Find an arc length of the current position in the base frame
   (localization).
3: for each path index $i$ do
4:     Initialize the maneuver in curvilinear coordinates.
5:     Generate a path with index $i$.
6:     Check the collision and arc length of the base frame
       without collision.
7: end for
8: for each path index $i$ do
9:     Calculate the cost for each path.
10:    if a path has no collision then
11:        Add the path to the set of path candidates.
12:    end if
13: end for
14: Sort the set of path candidates with respect to the cost.
15: if the set of path candidates is not empty then
16:     Choose the best path in the set of path candidates.
17: else
18:     Choose the best path with the longest arc length of the
        base frame without collision.
19: end if.
```

For collision detection, the obstacle map is checked for a path by considering the vehicle size and orientation (line 6). The cost for each generated path is then computed by considering the safety, smoothness, and consistency of the path (line 9). To prevent the selection of a path with a collision, the paths without a collision are simply added to the set of path candidates (line 11). In line 14, the set of path candidates with the lowest values of the computed cost in line 9 are sorted. In a normal situation, at least one path without collisions can be found in the set of path candidates (line 16). However, all path candidates may have a collision with obstacles in an uncertain environment. In this case, the path with the longest arc length of the base frame without a collision is selected (line 18). Although line 18 is not perfect for handling all exceptional situations (such as a completely blocked road), it is a sufficient treatment, assuming that the road along the base frame is not blocked. Line 18 was particularly used for the AVC and is not required for the handling of a blocked road.

E. Effects of Costs

Fig. 16 illustrates the simulation results for the presentation of the effects of the costs. In Fig. 16(a), the cost for safety is considered only for the selection of a path. The shape of the path shows a sharp turn that converges to the center of the road according to the safety cost, as shown in Fig. 16(b). Weighting for the safety cost can improve the performance of tracking the center of road, whereas it may cause sensitivity to the result of obstacle detection. In Fig. 16(c) and (d), the selection of the path that depends only on the smoothness cost and the smoothness distribution for path candidates are presented, respectively, at the same conditions in Fig. 16(a). The selected path in Fig. 16(c) is more straightforward than the selected path in Fig. 16(a).

However, the selection of a path that solely relies on the smoothness cost may generate a path that is very close to an obstacle, as shown in Fig. 16(c). Thus, the safety and the smoothness costs have to simultaneously be taken into account. Fig. 16(f) presents the distribution of the weighted sum of the safety and the smoothness costs with respect to the index of the path candidates. The result in Fig. 16(e) shows a reasonable path using the cost in Fig. 16(f).



(a)

(b)

(c)

(d)

(e)

(f)

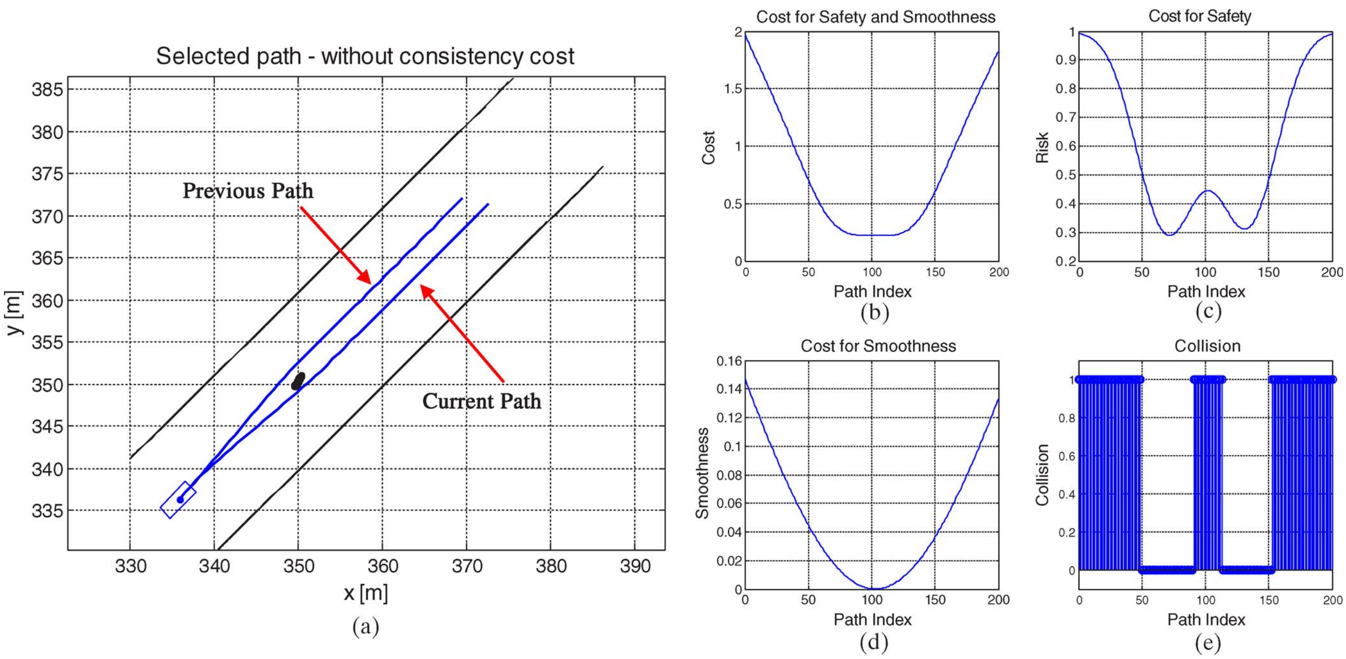
Fig. 16. Effects of costs for path selection.

Figs. 17 and 18 present the effect of the consistency cost for the selection of a path. The smoothness and safety of a path are taken into account for the selection of a path in Fig. 17, whereas the consistency of a path is additionally considered in Fig. 18. In Fig. 17(a), a current path is generated at the right side of an obstacle at the center of the road, whereas a previous path is generated at the left side of an obstacle. The reason for this change of path comes from the multiple minima of the weighted sum of the smoothness and safety costs, as shown in Fig. 17(b)–(e). Fig. 17(b)–(e) shows the weighted sum of costs, safety cost distribution, smoothness cost distribution, and collision checking data, respectively.

In Fig. 17(b), the weighted sum of smoothness and safety has a minimum value at around path index 100, and the shape of the cost sum is very flat in the neighborhood of the minimum value. This condition can increase the probability of the multiple minima of the total cost. On the other hand, the consistency cost is additionally considered for the selection of the path in Fig. 18. Fig. 18(b)–(e) shows the weighted sum of costs, safety cost distribution, smoothness cost distribution, and consistency cost distribution, respectively. In particular, Fig. 18(b) and (e) presents how the consistency cost works to treat the same situation as shown in Fig. 17. The total cost of the path in Fig. 18(b) has no slender shape in the neighborhood of the minimal point because of the consistency cost in Fig. 18(e). The selected path with the consistency cost in Fig. 18(a) shows that the path planner maintains a similar path to the previous path, without an abrupt change of direction.

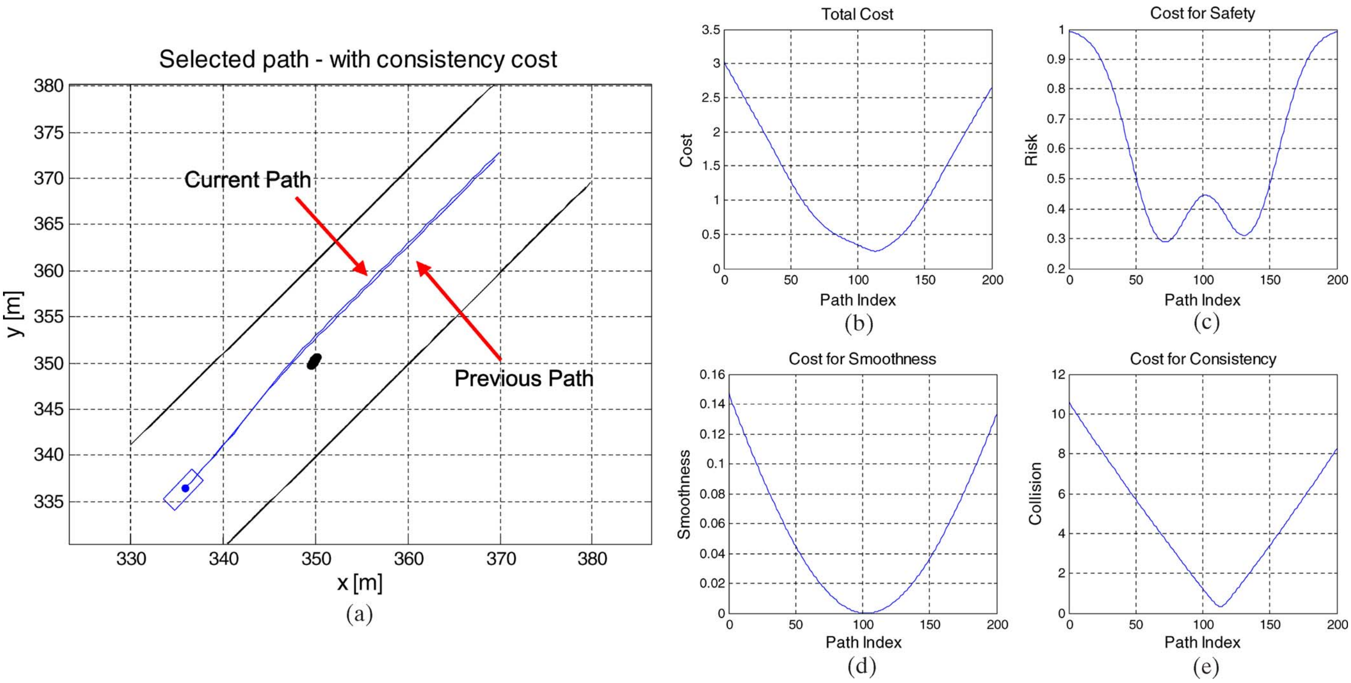
VI. Determination of the Target Speed

Vehicle speed has a significant role in the driving safety of autonomous vehicles. Therefore, multiple criteria were considered in determining the target vehicle speed for the



(a) (b) (c) (d) (e)

Fig. 17. Path selection without the consistency cost.



(a) (b) (c) (d) (e)

Fig. 18. Path selection with the consistency cost.

implementation of the autonomous vehicle A1, as shown in Fig. 19.

A. Speed Constraint for the Maximum Curvature of a Selected Path

First, the predefined speed limit of the road can easily be configured through the predefined waypoints list. However, the path planner in uncertain environments may generate a path that requires an abrupt change in the steering angle and a huge amount of steering angle. These situations can significantly affect the tracking performance and the stability of the vehicle control. In particular, the vehicle speed and the curvature of the path are closely coupled to the lateral acceleration of the vehicle motion, which is directly linked to the stability. The steering delay in high-speed driving also causes a performance degradation of the path tracking. For this reason, the lateral acceleration should be limited. The limit of the lateral acceleration $|a_y|_{\max}$ is a design factor of target vehicle speed. For the conservative design of the speed limit, the maximum curvature along the

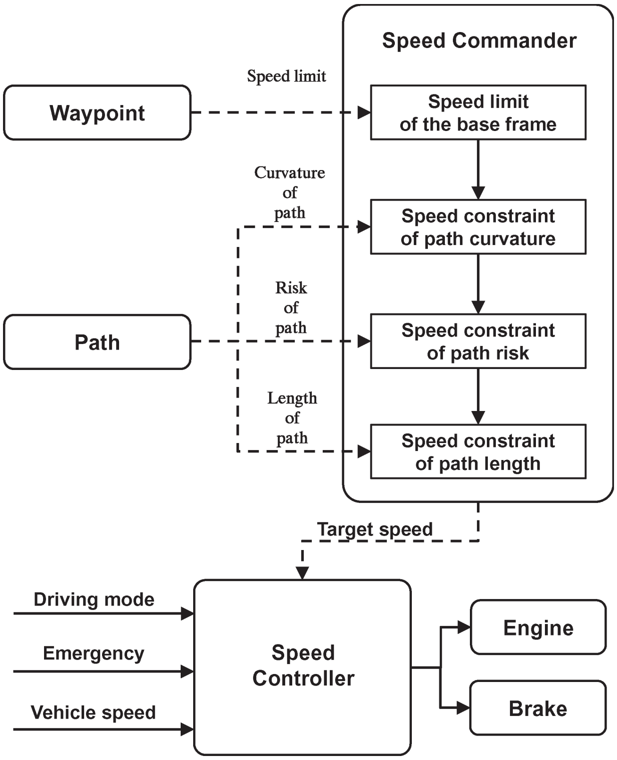


Fig. 19. Strategy of speed command.

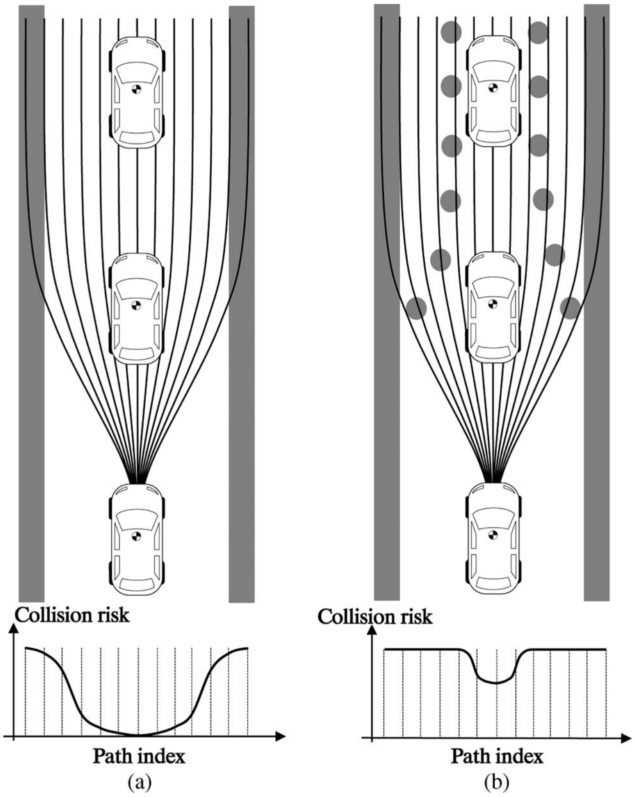


Fig. 20. Drivable road and collision risk of paths.

generated path is used to compute the speed limit. The vehicle speed limit that was computed by the maximum curvature of a path is given by

$$v_\kappa = \sqrt{\frac{|a_y|_{\max}}{\max \kappa(\text{path})}}. \tag{19}$$

B. Speed Adjustment by the Risk of a Selected Path

Generally, a driver cannot easily drive on a narrow road at high speed or adjacent to an obstacle. Similarly, autonomous driving requires a speed reduction on a narrow road to improve the accuracy of the vehicle control. To drive more safely, the target vehicle speed should be determined by considering the environment information, such as the proximity of obstacles and collision probabilities. The collision risk of paths provides how many obstacles exist adjacent to the selected path, as shown in Fig. 20. In Fig. 20(a), the wide road boundary and the absence of obstacle provide a low level of collision risk of the path. On the other hand, a narrowed road with obstacles increases the collision risk of the path, as shown in Fig. 20(b). For this reason, the speed commander adjusts an appropriate target speed based on the collision risk of the path, as shown in Fig. 21. The adjusted target vehicle speed is given by

$$v_{\text{safe}} = \left(1 - k_{\text{safe}} \cdot C_S^2[i]\right) v_{\text{curv}} \tag{20}$$

where k_{safe} is the safety gain for speed adjustment. The safety gain for speed adjustment is a heuristic design parameter that was experimentally chosen according to the driving style.

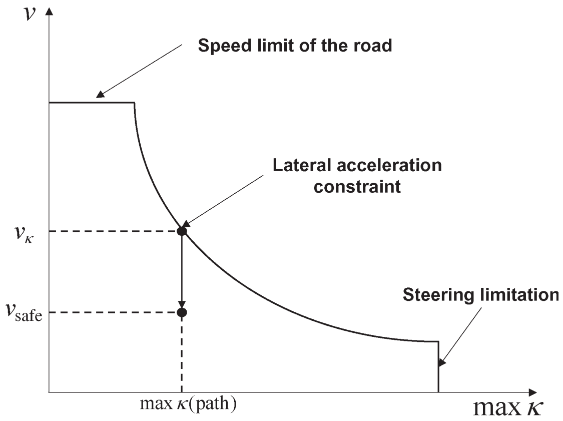


Fig. 21. Speed adjustment by the maximum curvature and the risk of the selected path.

C. Speed Constraint for the Length of the Selected Path

During autonomous driving, several exceptional driving situations occurred. In this case, the path planner generates the reduced path, which is caused by the absence of path candidates for driving without a collision. This situation is an abnormal driving condition for the autonomous vehicle. Therefore, the exception-handling function is also required for the speed commander and the path planner. To handle the exceptional case, the length of the path is important information that provides the distance from the collision. Once the length of the path is reduced, the mode of the speed commander is changed to the emergency mode, which decreases the target vehicle speed to the preprogrammed condition.

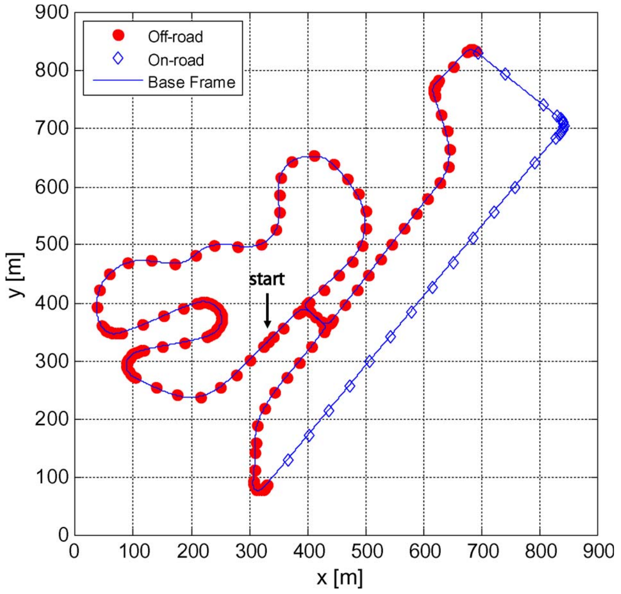


Fig. 22. Race circuit and waypoints of the AVC.

TABLE II
MISSIONS FOR THE AVC

Mission	Penalty
Lane Keeping	3 min
Speed Limit	5 min
Stop at crosswalk line	5 min
Tunnel	3 min
One obstacle avoidance	3 min
Gate	3 min
Median strip	3 min
Haystack	3 min
Two obstacles	3 min
Narrow road	3 min

VII. Experiments

A. Autonomous Vehicle A1 for the AVC in Korea

On November 5, 2010, the Hyundai–Kia Automotive Group held the AVC, which was the first race of autonomous vehicles in Korea. The goal of an autonomous vehicle in the AVC was to drive the circuit shown in Fig. 22 and to complete the missions in Table II without human intervention. The maximum curvature of the road was 0.1187. The winner was selected by combining the travel time and the penalties for each mission. The driving missions were performed both on a paved road and off-road. The off-road driving duties involved a tunnel, one obstacle, a gate, a median strip, a haystack, two obstacles, and a narrow road. The remainder of the missions was carried out on the paved road.

The course for the 2010 AVC consisted of 3 km of off-road and 1 km of paved road. A1 finished the course in 7 min 40 s, completing all of the missions and winning the AVC. In Fig. 22, the red dots denote the waypoints of the off-road of the 2010 AVC track, and the blue diamond represents the waypoint of the paved road of the AVC track.

The blue line in the figure is the base frame that is generated by the cubic spline curve using the given waypoints. This base frame is computed before the race for the path planning of the A1. The path-planning algorithm does not directly use the base frame as a path but, instead, uses it as a horizontal axis of the curvilinear coordinate system for obstacle avoidance.

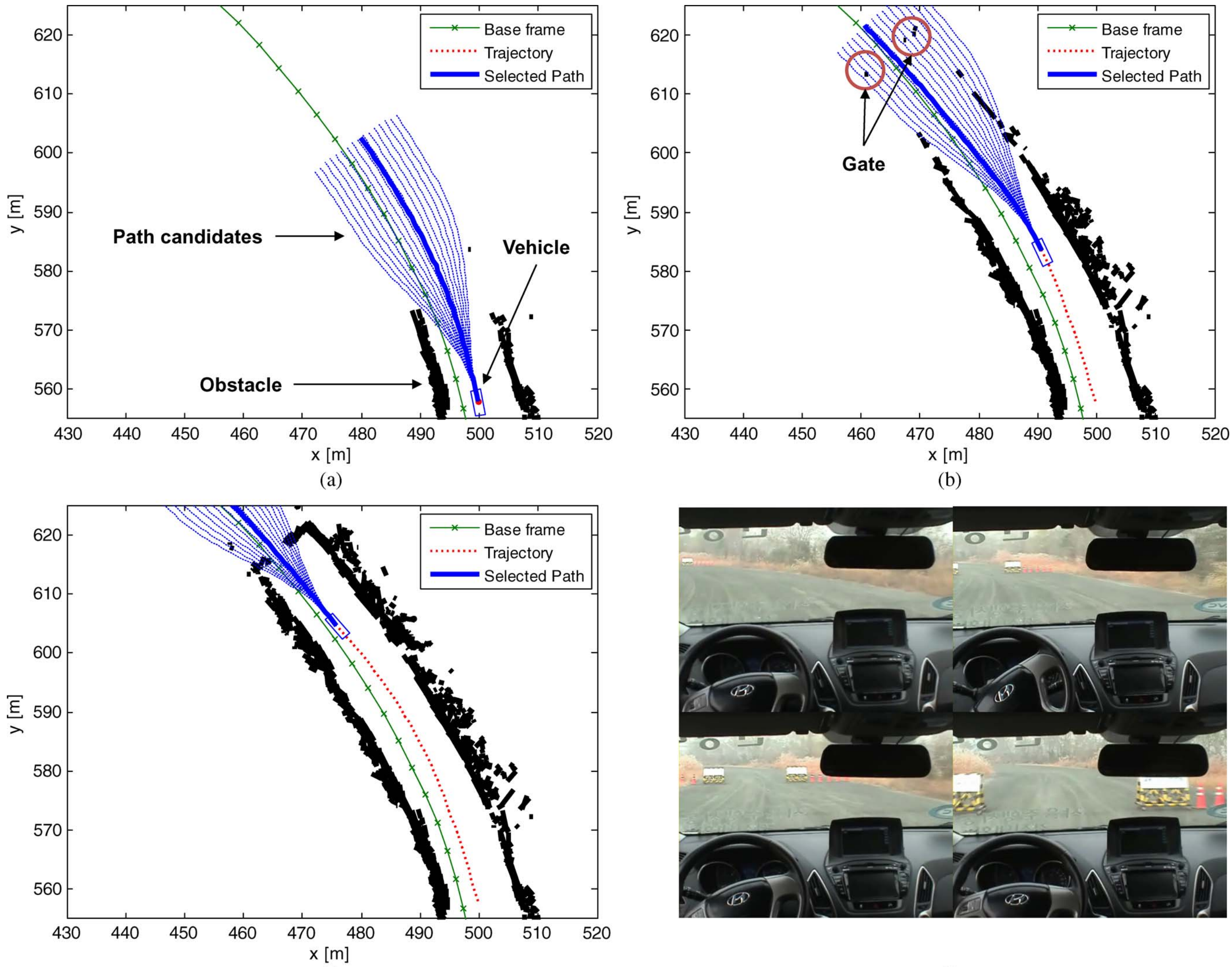
B. Experimental Results

In this section, several results of the autonomous driving in the AVC are presented. The logged data that were obtained during the autonomous driving of A1 at the gate mission of the AVC course are shown in Fig. 23. The black lines represent obstacle data that were collected and filtered during the racing, and the green lines represent the base frame that was generated using the given waypoints. The dashed blue lines indicate the path candidates, whereas the solid blue line is the selected path from the path candidates. Fig. 3(a)–(c) shows the generation of the path candidates and the selection of a path while A1 is passing the gate. Snapshots of the video that were taken in A1 during the AVC race are presented in Fig. 23(d).

In Fig. 23(a), the gate was not detected, and the path that slowly drifts apart from the base frame was selected. In this case, the smoothness cost plays an important role in selecting a smooth path from the path candidates. The result of tracking the path in Fig. 23(a) is the trajectory of the vehicle, presented as red dots in Fig. 23(b). A situation in which the gate is detected at about 40 m ahead is shown in Fig. 23(b). To avoid collision with the gate, the path was selected with a change in the lateral offset to the left. The change in the lateral offset causes more left turn of the steering, as shown in Fig. 23(d). Because A1 steered from a distance in Fig. 23(b), it easily passed the gate without abrupt steering, as shown in Fig. 23(c). The trajectories of A1 in Fig. 23(c) show that the vehicle position and orientation changed to allow A1 to safely pass the gate.

Fig. 23 shows that the actual base frame is not the center line of the road. Nevertheless, A1 safely drives the course, because this base frame was only used as a curvilinear coordinate system for the generation of path candidates.

Two obstacles in Table II require complex maneuvering of the vehicle instead of a simple shift in the lateral offset. A mission that consists of two obstacles that are sequentially placed at a distance of 25 m on different sides of the road is shown in Fig. 24. Because this mission requires a snake-like motion of the vehicle to avoid the obstacles, a path with a fixed arc length of the base frame cannot avoid a collision. Therefore, if collisions are detected for all path candidates that simply shift the lateral offset in the curvilinear coordinates, only the part of the paths that are collision-free is used, and the length of the path is reduced. In Fig. 24(a), A1 approaches two obstacles with the selected path, which causes A1 to shift to the right side of the road. As a result of the path in Fig. 24(a), all of the path candidates in Fig. 24(b) have a collision with a fixed arc length of the base frame. To address this situation, the path with the



(a)

(b)

(c)

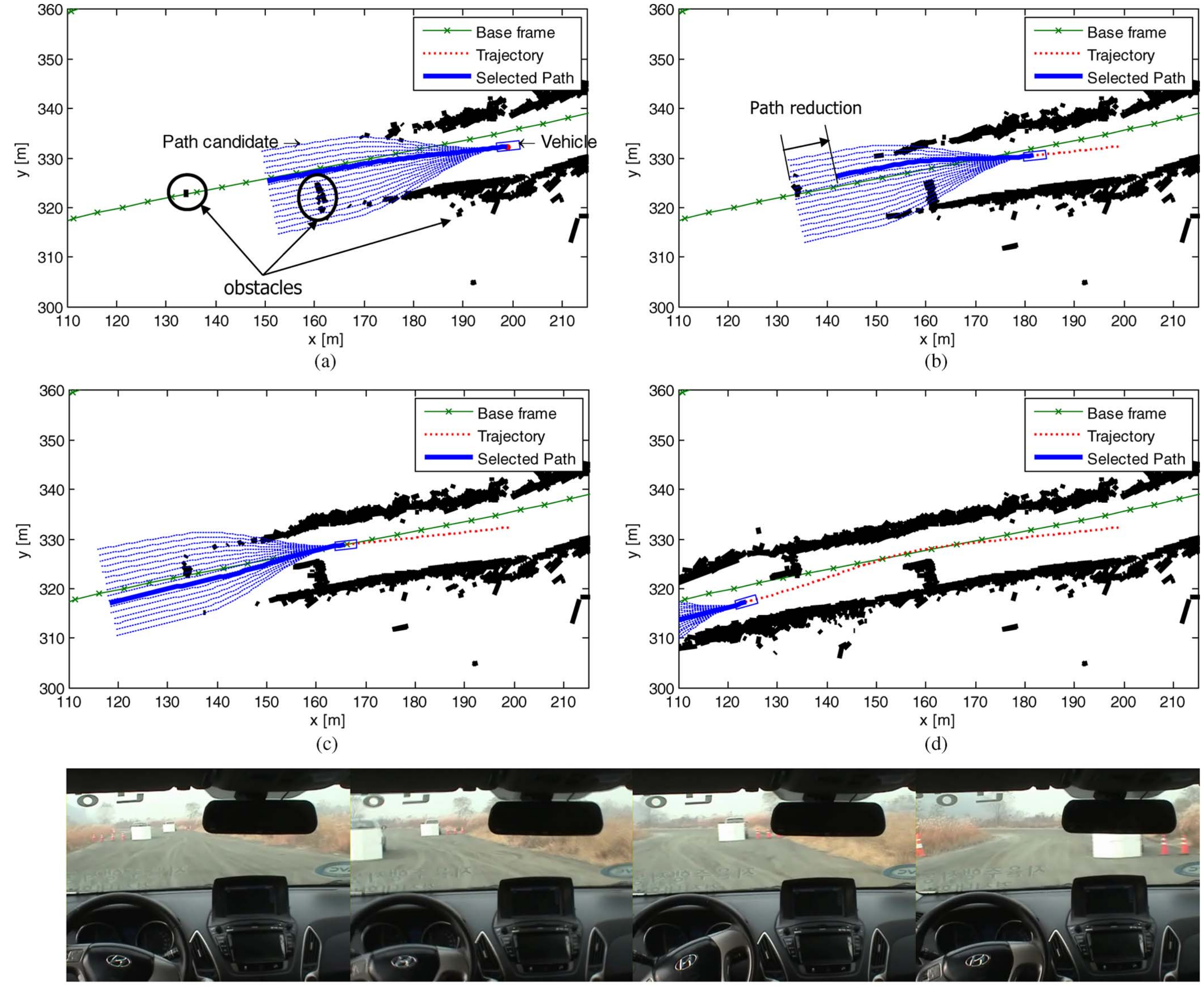
(d)

Fig. 23. A1 driving at the gate.

longest arc length of the base frame without a collision, based on line 18 of Algorithm 1, is chosen from the path candidates. A collision check was performed by considering an extension of the vehicle size. For this reason, the length of the selected path in Fig. 24(b) is reduced. When the path reduction was detected, A1 slowed down its speed with respect to the length of the path. The speed adjustment according to the length of the path enabled A1 to pass the two obstacles safely. After progressing from the situation in Fig. 24(b), A1 naturally found a path without a length reduction, as shown in Fig. 24(c). The path in Fig. 24(c) makes it possible for A1 to avoid two obstacles. The resulting trajectory that avoids two obstacles is shown as red dots in Fig. 24(d). The shape of the trajectory shows a smooth change in the maneuvering of A1 from the center of the road to the right and left sides of the road. Snapshots of the video taken in A1 during the two obstacles mission are displayed in Fig. 24(e).

The start and the finish of the race by the A1 at the AVC are shown in Fig. 25. At the AVC, the autonomous vehicle should turn right at the first corner, as shown in Fig. 25(a). The base frame guides A1 to turn right at the first corner after starting from the start line, as shown in Fig. 25(b). Although the base frame in Fig. 25(a) is very close to obstacles, the trajectory of A1 is not close to obstacles. Because the base frame is used only as an axis of the curvilinear coordinate system, the trajectory does not collide with the obstacles. At every time step, a path is selected from the path candidates by the cost function that considers the safety of the path and the smoothness of the path. As a result of the frequent path generation with the safety and smoothness costs, the trajectory of A1 is sufficiently smooth and is not unnecessarily close to obstacles, as shown in Fig. 25(a). The shape of the trajectory demonstrates the effect of the smoothness cost. In Fig. 25(a), A1 entered the corner on the inside and exited from the corner on the outside edge of the track.

The logged data at the finish line of the race are shown in Fig. 25(c). The final mission on a narrow road was performed in front of the finish line. Only the right side of the road was drivable, as shown in Fig. 25(d). A1 detected the obstacle early and found a smooth path in Fig. 25(c). No collision with an obstacle occurred while A1 was driving the AVC course.



(a) (b) (c) (d) (e)

Fig. 24. Avoiding two obstacles.

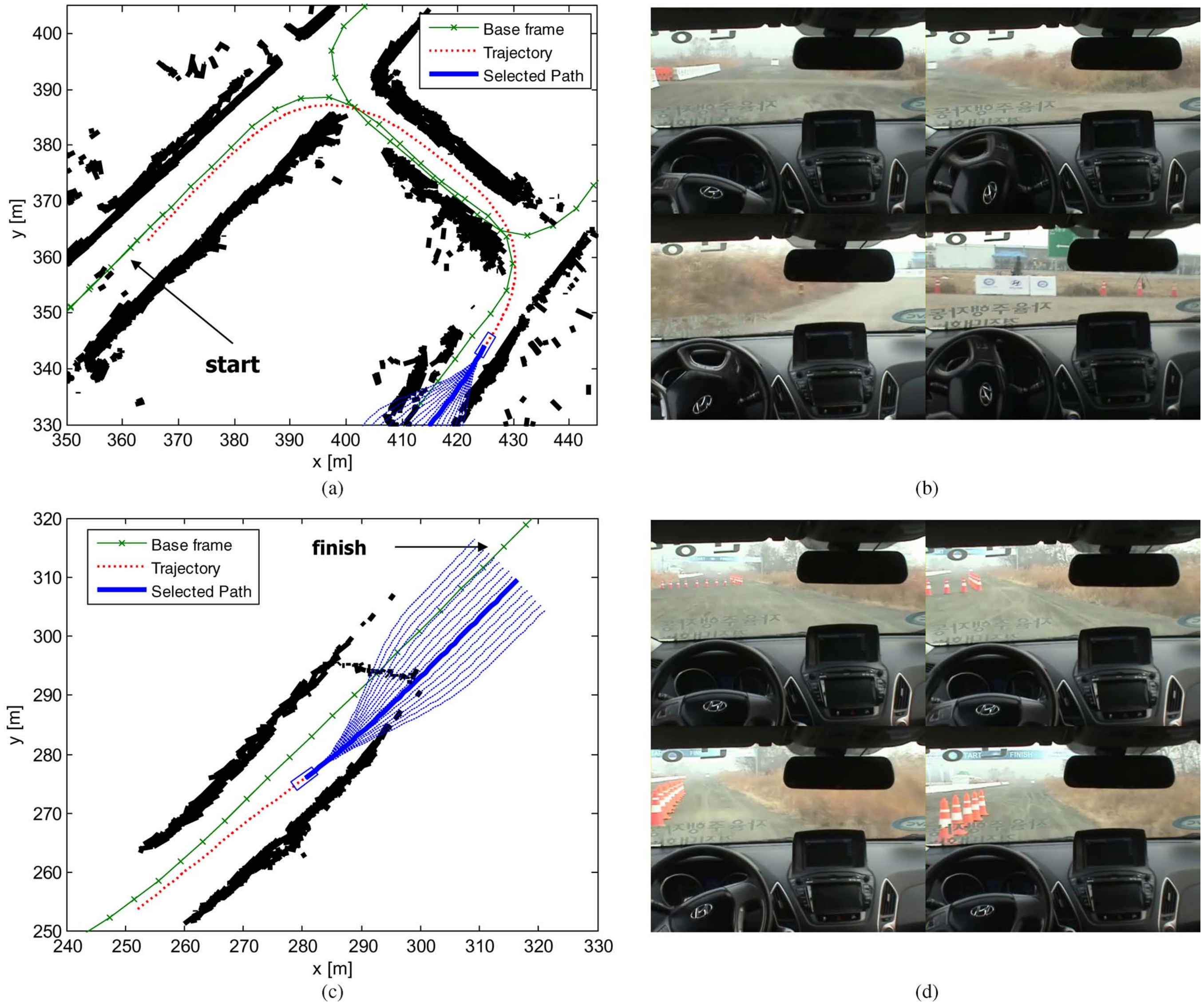
VIII. Conclusion

In this paper, a local path-planning algorithm for off-road autonomous driving with static obstacles has been described.

Our path-planning strategy provides a design framework for local path planning that uses the global route and local environmental data. The proposed planning algorithm utilizes directional information from the global route given by predefined waypoints. The directional information of the global route is blended with the maneuvering of the vehicle motion by adjusting the lateral offset of the base frame in a curvilinear coordinate system.

For the selection of a path, the safety, smoothness, and consistency costs are adopted to evaluate the effects of both the uncertainty of the environment and the vehicle dynamics. The uncertainty of the environment is treated in the form of the safety cost of the path candidates, which yields computational efficiency and prevents the autonomous vehicle from unnecessarily passing close to obstacles. With regard to the vehicle dynamics, the smoothness cost penalizes a path that has a sudden curve, whereas the consistency cost for a path reduces a sudden change in the path. The selection of a path was characterized by a combination of the costs and can intuitively be tuned by the design criteria in the proposed path-planning scheme.

In this paper, we have focused on the description of the local path planning for autonomous vehicles in a practical way, with experiments rather than through theoretical improvement of the path-planning algorithm. Although most of the concepts applied in this paper, such as the utilization of curvilinear coordinates and cost functions for path selection, are individually not novel, the integration of these concepts presents successful results in the experiments. The strategy of path planning that has been presented here was effectively implemented and validated in the static obstacles by the autonomous vehicle A1, which won the 2010 AVC.



(a) (b) (c) (d)

Fig. 25. Start and finish of the AVC track.

Although dynamic objects such as moving vehicles and humans are not considered in this paper, the planning algorithm can be extended to manage the dynamic environment with information on dynamic objects.

REFERENCES

[1] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouvelière, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 589–606, Sep. 2010.

[2] Ü. Özgüner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience," *Proc. IEEE*, vol. 95, no. 2, pp. 397–412, Feb. 2007.

[3] J. Hwang, D. Lee, K. Huh, H. Na, and H. Kang, "Development of a path planning system using mean shift algorithm for driver assistance," *Int. J. Autom. Technol.*, vol. 12, no. 1, pp. 119–124, Feb. 2011.

[4] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The Great Robot Race*. New York: Springer-Verlag, 2007.

[5] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. New York: Springer-Verlag, 2010.

[6] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, Sep. 2006.

[7] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y. W. Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, Aug. 2008.

[8] S. Waydo and R. M. Murray, "Vehicle motion planning using stream functions," in *Proc. Int. Conf. Robot. Autom.*, 2003, pp. 2484–2491.

[9] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA: MIT Press, 2005.

[10] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semistructured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, Apr. 2010.

[11] D. Ferguson and A. Stentz, "The field d* algorithm for improved path planning and replanning in uniform and nonuniform cost environments," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-19, 2005.

[12] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Robot.*, vol. 26, no. 3, pp. 308–333, Mar. 2009.

[13] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using RRT," in *Proc. Int. Conf. Intell. Robots Syst.*, 2008, pp. 1681–1686.

[14] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[15] S. M. LaValle and J. J. Kuffner, Jr, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[16] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Sep. 2009.

[17] M. Montemerlo, J. Becker, S. Shat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, Sep. 2008.

[18] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét frame," in *Proc. ICRA*, 2010, pp. 987–993.

[19] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *J. Field Robot.*, vol. 25, no. 11/12, pp. 939–960, 2008.

[20] K. Jo, K. Chu, K. Lee, and M. Sunwoo, "Integration of multiple vehicle models with an IMM filter for vehicle localization," in *Proc. IEEE IV Symp.*, San Diego, CA, 2010, pp. 746–751.

[21] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 329–343, Mar. 2012.

[22] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 971–985, Mar. 2012.

[23] H. Wang, J. Kearney, and K. Atkinson, "Arc-length parameterized spline curves for real-time simulation," in *Proc. 5th Int. Conf. Curves Surf.*, 2002, pp. 387–396.

[24] B. Guenter and R. Parent, "Computing the arc length of parametric curves," *IEEE Comput. Graph. Appl.*, vol. 10, no. 3, pp. 72–78, May 1990.

[25] I. N. Bronshtein, K. A. Semendyayev, and K. A. Hirsch, *Handbook of Mathematics*. Berlin, Germany: Springer-Verlag, 2007.

[26] H. Wang, J. Kearney, and K. Atkinson, "Robust and efficient computation of the closest point on a spline curve," in *Proc. 5th Int. Conf. Curves Surf.*, 2002, pp. 397–406.

[27] T. Barfoot and C. Clark, "Motion planning for formations of mobile robots," *Robot. Auton. Syst.*, vol. 46, no. 2, pp. 65–78, Feb. 2004.

[28] B. Siciliano, L. Sciavicco, and L. Villani, *Robotics: Modeling, Planning and Control*. New York: Springer-Verlag, 2009.

[29] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ: Wiley, 2006.

[30] D. Ferguson, C. Baker, M. Likhachev, and J. Dolan, "A reasoning framework for autonomous urban driving," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 775–780.

[31] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments—Part II," in *Proc. Int. Conf. Intell. Robots Syst.*, 2008, pp. 1070–1076.

[32] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments—Part I," in *Proc. Int. Conf. Intell. Robots Syst.*, 2008, pp. 1063–1069.

[33] D. Ferguson, M. Darms, C. Urmson, and S. Kolski, "Detection, prediction, and avoidance of dynamic obstacles in urban environments," in *Proc. IEEE Intell. Veh. Symp.*, Eindhoven, The Netherlands, 2008, pp. 1149–1154.

[34] J. Daniel, A. Birouche, J. P. Lauffenburger, and M. Basset, "Energy constrained trajectory generation for ADAS," in *Proc. Int. Intell. Veh. Symp.*, 2010, pp. 244–249.

[35] H. Delingette, M. Hebert, and K. Ikeuchi, "Trajectory generation with curvature constraint based on energy minimization," in *Proc. Int. Workshop Intell. Veh. Symp.*, 1992, pp. 206–221.

[36] Y. J. Kanayama and B. I. Hartman, "Smooth local-path planning for autonomous vehicles," *Int. J. Robot. Res.*, vol. 16, no. 1, pp. 263–284, Feb. 1997.

[37] B. K. P. Horn, "Curve of least energy," *ACM Trans. Math. Softw.*, vol. 9, no. 4, pp. 441–460, Dec. 1983.

Keonyup Chu (S'10) received the B.S. degree in mechanical engineering and the M.S. and Ph.D. degree in automotive engineering from Hanyang University, Seoul, Korea, in 2004, 2006, and 2011, respectively.

He is currently with the Autonomous Control and Electronics Laboratory, Hanyang University. His research activities include the design of intelligent vehicles, hard real-time systems, distributed control systems, and in-vehicle networks. His current research interests include the motion planning and collision mitigation of vehicles. He has also worked on the development of autonomous vehicles.

Minchae Lee (S'10) received the B.S. degree in electronics and computer engineering and the M.S. degree in automotive engineering in 2006 and 2008, respectively, from Hanyang University, Seoul, Korea, where he is currently working toward the Ph.D. degree with the Automotive Control and Electronics Laboratory.

His research activities include the design of intelligent vehicles, distributed control systems, in-vehicle networks, and automotive software engineering. His research interests include lane detection, object detection, pattern recognition, and the engineering applications of evolutionary techniques.

Myoungho Sunwoo (M'81) received the B.S. degree in electrical engineering from Hanyang University, Seoul, Korea, in 1979, the M.S. degree in electrical engineering from the University of Texas, Austin, in 1983, and the Ph.D. degree in system engineering from Oakland University, Rochester, MI, in 1990.

In 1985, he joined General Motors Research (GMR) Laboratories, Warren, MI. He has been working in the area of Automotive Electronics and Control for 26 years. During his nine-year tenure with GMR, he worked on the design and development of various electronic control systems for powertrain and chassis. Since 1993, he has led research activities as a Professor with the Department of Automotive Engineering, Hanyang University, which is one of the largest engineering schools in Korea. His work has been focused on automotive electronics and control such as the modeling and control of internal combustion engines, the design of automotive distributed real-time control systems, intelligent autonomous vehicles, and automotive education programs. During his professional career, he has published 55 international journal articles and 68 international conference proceedings. He is the holder of 19 patents. In addition, he successfully completed more than 50 research projects with the Korean Government and automotive companies such as Hyundai, Kia, Mando, Hyundai MOBIS, and Freescale. His laboratory, i.e., the Automotive Control and Electronics Laboratory (ACE Lab), has been selected as a National Research Laboratory by the Korean Government because of its outstanding research accomplishments. His autonomous vehicle called "A1" won the First National Autonomous Vehicle Competition, which was organized by Hyundai Motor Company hosted in Korea in November 2010. He has continuously consulted for the Korean Government and automotive industries.

Dr. Sunwoo is currently an Academician of the National Academy of Engineering of Korea and the Chairman of the Steering Committee for the Green Car Strategy Forum of the Ministry of Knowledge Economy. In recognition of his distinguished achievements, he has received notable awards such as the Grand Award of Academic–Industrial Cooperation from the Korean Academic–Industrial Foundation, the Best Scientist/Engineer Award of the month from the Korean Ministry of Science and Technology, and the Award of Technology Innovation from the Prime Minister of Korea.