

# Co-Pilot Agent for Vehicle/Driver Cooperative and Autonomous Driving

Samyeul Noh, Byungjae Park, Kyoungwan An, Yongbon Koo, and Wooyong Han

ETRI's Co-Pilot project is aimed at the development of an automated vehicle that cooperates with a driver and interacts with other vehicles on the road while obeying traffic rules without collisions. This paper presents a core block within the Co-Pilot system; the block is named "Co-Pilot agent" and consists of several main modules, such as road map generation, decision-making, and trajectory generation. The road map generation builds road map data to provide enhanced and detailed map data. The decision-making, designed to serve situation assessment and behavior planning, evaluates a collision risk of traffic situations and determines maneuvers to follow a global path as well as to avoid collisions. The trajectory generation generates a trajectory to achieve the given maneuver by the decision-making module. The system is implemented in an open-source robot operating system to provide a reusable, hardware-independent software platform; it is then tested on a closed road with other vehicles in several scenarios similar to real road environments to verify that it works properly for cooperative driving with a driver and automated driving.

**Keywords:** Cooperative driving, automated driving, enhanced map, decision-making, trajectory generation.

## I. Introduction

A self-driving car is a vehicle that can drive from a current position to a destination without human intervention by recognizing driving environments, determining the behaviors for various driving situations, and controlling actuators to move the car. Nowadays, self-driving cars are being focused upon and are considered as the future of personal transportation since they can reduce those traffic accidents that are a result of human error, save energy, give drivers greater convenience, and help people who are incapable of driving.

However, there are limitations and barriers to deploying self-driving cars on actual roads owing to several technical and legal issues [1]. Current state-of-the-art technologies cannot guarantee the performance of localization, object recognition, and decision-making for all possible and complicated urban environments. In addition, there are currently no well-defined rules or laws for self-driving cars on real roads.

To cope with the technical and legal issues described above, the proposed system supports a highly automated level of automation. The automation levels can be divided into the following five categories [2]–[3]: driver-only, assisted, partially automated, highly automated, and fully autonomous. At the highly automated level, the driver should take over the control within a certain time constraint when the system requests it. If a take-over request is denied, then the system stops or pulls over the car safely. We call the proposed system Co-Pilot, since it acts like a copilot of an airplane helping or replacing the pilot.

The Co-Pilot system has a unique characteristic that makes it stand out from other research vehicles [4]–[6] (that is, its highly automated level). First, the system can change or recommend control authority between a driver and the system. In addition, it can help a driver in the longitudinal or lateral directions or

---

Manuscript received Aug. 8, 2014; accepted July 16, 2015.

This work was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0101-15-0134, Development of Decision Making/Control Technology of Vehicle/Driver Cooperative Autonomous Driving System based on ICT).

Samyeul Noh (corresponding author, samuel@etri.re.kr), Byungjae Park (bjp@etri.re.kr), Kyoungwan An (mobileguru@etri.re.kr), Yongbon Koo (ybkoo@etri.re.kr), and Wooyong Han (wyhan@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

drive in a highly automated manner. Second, the system is more practical and applicable to real driving situations. In other words, it considers not only normal scenarios, such as lane following or lane change, but also minimum risk maneuvers when a driver denies or cannot take over the control when the system requests to do so during automated driving. Finally, the system, designed as a reusable software platform, is operated based on an open-source robot operating system called ROS [7] and divided into several well-defined functional modules. In addition, each module is independent from hardware, such as sensors and vehicles, so that the system can be easily replaced with more advanced versions.

The remainder of this paper is organized as follows. Section II begins with an overall system architecture. Section III explains a road map model used throughout the system. Section IV presents a decision-making module based on situation assessment and behavior planning, and Section V continues with a trajectory generation module. Section VI provides experimental results through in-vehicle tests on real roads, and Section VII summarizes the contributions of this paper and describes future work.

## II. Overview of Co-Pilot System

### 1. Hardware System

The Co-Pilot system utilizes a 2012 Kia Sorrento R modified for cooperative driving between a driver and the system. Since it provides both driver and system control capability, it maintains normal controls (steering, throttle, braking, and shift gear) so that a driver can quickly and easily take over the control from the system. In addition, it has several sensors to recognize the environment and driver status — an eye tracker, a 3D laser scanner, a GPS/IMU. Figure 1 shows the hardware platform of the Co-Pilot system.

### 2. Software System

The Co-Pilot system consists of five blocks — driver status monitoring, driving environment perception, driving control authority decision, Co-Pilot agent, and vehicle motion control. Figure 2 shows the software architecture of the Co-Pilot system. This subsection briefly describes other blocks in the Co-Pilot system with the exception of the Co-Pilot agent block.

#### A. Driver Status Monitoring Block

This block is composed of two modules — direct recognition and indirect recognition. The indirect recognition module infers the driver's status (normal, drowsy, and sleepy) from the vehicle's internal information (steering, throttle, and

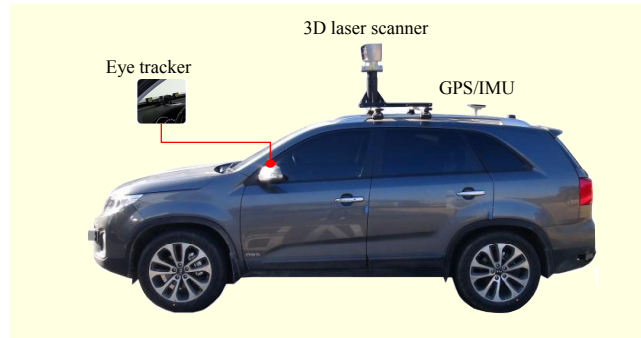


Fig. 1. Co-Pilot hardware platform. Co-Pilot system is equipped with eight different sensors — an eye tracker, a 3D laser scanner, and a GPS/IMU.

brake). The direct recognition module recognizes the driver's status by tracking the driver's eyelids and eye gaze with an eye tracker.

#### B. Driving Environment Perception Block

This block is composed of four modules — static/dynamic object detection and tracking, drivable semantic map, lane recognition, and traffic signal recognition. However, it only functions object detection for now by using a 3D laser scanner. The object detection data are sent to the Co-Pilot agent block to make a decision and generate a trajectory.

#### C. Decision of Driving Control Authority Block

This block determines the vehicle state transition for cooperative driving between the driver and system based on three values — the estimated risk value of the driver driving from the driver status monitoring block, the inferred value of the driver intervention from the vehicle motion control block, and the direct value of HVI devices by the driver's input.

#### D. Vehicle Motion Control Block

This block is directly connected to the vehicle's actuators and controls the vehicle motion based on the trajectories generated by the Co-Pilot agent block. The generated trajectories are composed of waypoints; however, the control inputs of this block consist of the values of throttle, brake, and steering.

### 3. Co-Pilot Agent Block

The Co-Pilot agent block is the core part of the Co-Pilot system, which includes four modules — road map, global pose, decision-making, and trajectory generation. The road map module builds the road map data to provide enhanced and detailed map data to other modules. The map data include lane-level geometry data (such as centerline waypoints), lane-level topological data (such as nodes, links, and segments), and lane-

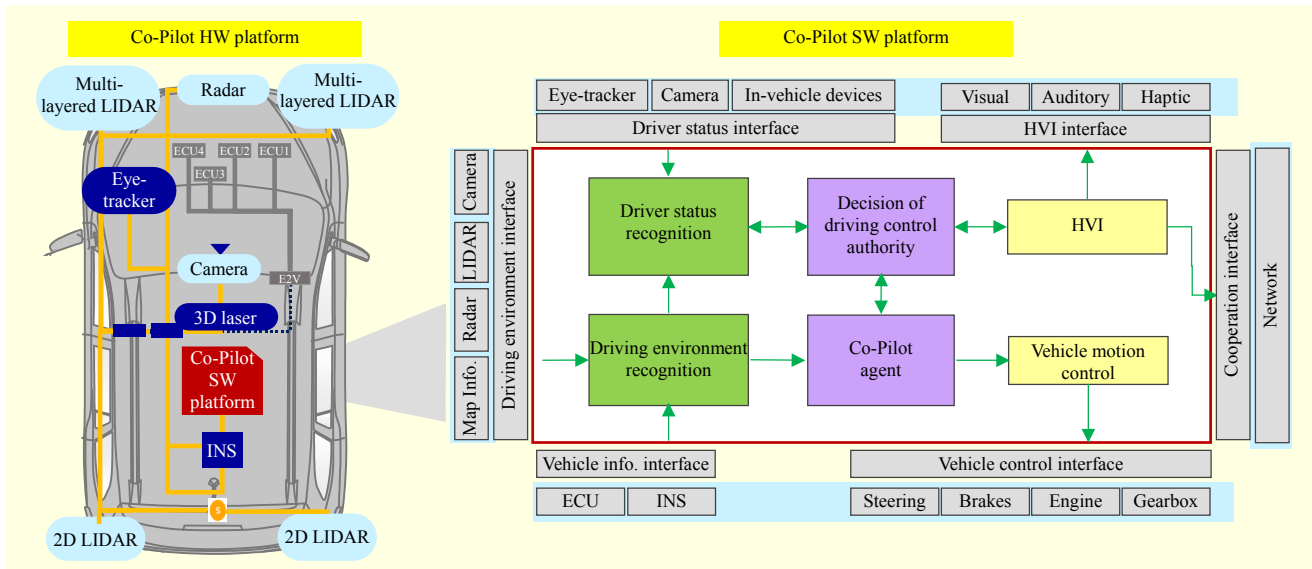


Fig. 2. Co-Pilot software architecture consisting of five blocks — driver status monitoring, driving environment perception, decision of driving control authority, Co-Pilot agent, and vehicle motion control.

level semantic data (such as road properties). The global pose module gathers the pose of the vehicle and the pose errors from the GPS/IMU sensor at a 100 Hz cycle. Its accuracy under good conditions is 2 cm [8]–[9], which is good enough to drive without any additional corrections. The decision-making module makes a maneuver decision for cooperative driving between a driver and the system by selecting the control modes, recognizing the driving situations, evaluating the collision risk of the situation, and planning behaviors not only to follow the global path but also to avoid colliding with other vehicles. The trajectory generation module generates a trajectory to achieve the maneuver given by the decision-making module and provides it to the vehicle motion control block.

### III. Road Map Generation

#### 1. Road Map Model

The road map data used for the vehicle control systems are different from those of car navigation systems in that they need more enhanced and detailed map features, such as lane-level geometries and attribute data. In this paper, we utilize the enhanced map data for several purposes — to filter out noisy sensor data by setting the region of interest through the map data; to recognize the map data as one of the reliable sensors that enables data fusion with sensor data streams for enhancing the reliability and enlarging the sensing coverage; to determine a reference route from the start to the destination by using the lane-level topological data in the global route planner module; to extract semantic information of the ego vehicle and other vehicles by combining them with the map data; and to give the

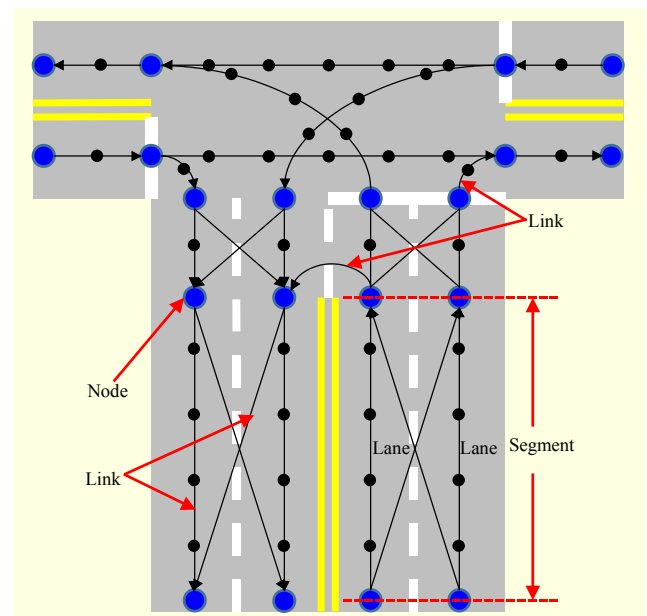


Fig. 3. Enhanced and detailed map data that include lane-level geometries and attribute data (such as node, lane, link, segment, and centerline waypoints).

road shape information to the trajectory generation module to use the lane-level geometric data as a reference path [10].

The road map model is defined as shown in Fig. 3. Nodes and links exist for each lane and have connections when there are legal paths that vehicles can follow. In addition, special links for lane change exist for the lane-level global route planning. The lanes are grouped as a segment to find the drivable space using the global route. The small black circles on the links represent the dense waypoints of centerlines of each lane that are used as a reference path for the trajectory

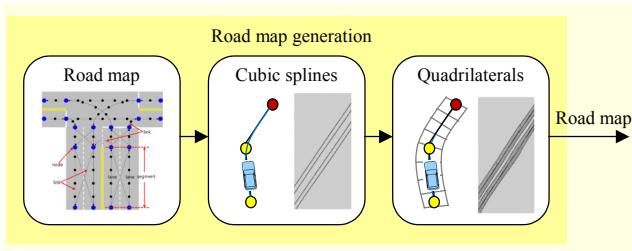


Fig. 4. Flow diagram of road map generation. Based on lane-level enhanced map data, quadrilaterals are provided to other modules by dividing each lane into adjacent quadrilaterals that tile drivable spaces of road.

generation module. The attributes not shown in Fig. 3 include lane numbers, lane widths, lane lengths, lane boundary information, speed limits, and desired speeds.

In addition, each lane is divided into adjacent quadrilaterals that tile the drivable space of the road, as shown in Fig. 4 [11]. The quadrilaterals have two main purposes — to provide a way of extracting semantic information of objects by classifying object data from the environmental perception as being in the current lane, in an adjacent lane, outside an adjacent lane, or off the road entirely; and to unify all data into one type of data, a quadrilateral. We use standard cubic splines with a Bezier basis to connect the discrete series of waypoints that define a lane. We express the curve in terms of third degree Bernstein polynomials, which are defined as:

$$c(u) = \sum_{i=0}^3 p_i \binom{3}{i} u^i (1-u)^{3-i}. \quad (1)$$

The  $p_i$  coefficients are Bezier control points. Using this curve with the lane widths defined in the map, we generate quadrilaterals that give the vehicle a rough guess regarding the shape of the roadway.

#### IV. Situation Assessment and Behavior Planning

This section details the decision-making module based on situation assessment and behavior planning. In order for the system to achieve the destination without collisions, it has to evaluate traffic situations and determine maneuvers every cycle. In this paper, we present risk-observer distribution-based situation assessment and behavior planning to produce collision-free maneuvers based on the results of each risk observer.

##### 1. Situation Assessment

This module, designed to recognize and evaluate the current traffic situation, divides the road region around the vehicle into six lane-level regions (that is, left forward, left backward, current forward, current backward, right forward, and right

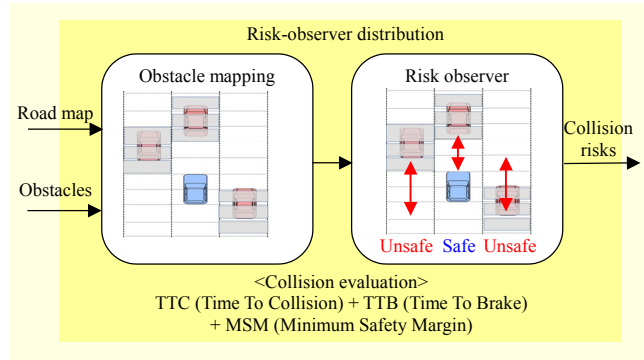


Fig. 5. Situation assessment based on risk-observer distribution. Collision risks for each risk observer are evaluated through three safety factors — TTC, TTB, and MSM.

backward). Six risk observers are assigned to the corresponding regions to integrate dynamic objects with the given road map as well as to evaluate the collision risk of the assigned regions (see Fig. 5). The object fusion with the map data functions not only to unify environmental perception data into a quadrilateral data type but also to combine the semantic information of the map data with the object tracking data. In addition, it filters out noisy data off the road. Each risk observer is composed of a subset of the quadrilateral list based on the road map data and reports the results of collision risk, which represents whether its assigned region is safe to enter.

To evaluate the collision risk of the current traffic situation, three safety factors are used as a threat measure. The first safety factor is time to collision (TTC), which is defined as the time required for two vehicles to collide if they continue at their respective current speeds and trajectories. The second safety factor is time to brake (TTB), which is defined as the time required for the collision to occur if the vehicle keeps its current speed and trajectory. The last safety factor is minimal safety margin (MSM), which is defined as the distance required for two vehicles to keep the minimal safety. Each safety factor is defined respectively as [12]:

$$TTC = \begin{cases} \frac{x_k^n - x_k^e}{v_k^e - v_k^n} & \text{if } \frac{x_k^n - x_k^e}{v_k^e - v_k^n} > 0, \\ +\infty & \text{otherwise,} \end{cases} \quad (2)$$

$$TTB = \frac{x_k^n - x_k^e}{v_k^e}, \quad (3)$$

$$MSM = |x_k^n - x_k^e|, \quad (4)$$

where  $x_k^n$  and  $v_k^n$  are a distance and a velocity of the nearest object in each assigned region at time  $k$ , respectively. Variables  $x_k^e$  and  $v_k^e$  are a distance and a velocity of the ego vehicle at time  $k$ , respectively.

With regard to these safety factors, we define each risk

parameter corresponding to TTC, TTB, and MSM as [12]:

$$R_{\text{TTC}} = 1_A(\text{TTC}) = \begin{cases} 1 & \text{if } t_k^n < \bar{t}^{\text{ttc}}, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

$$R_{\text{TTB}} = 1_A(\text{TTB}) = \begin{cases} 1 & \text{if } t_k^n < \bar{t}^{\text{ttb}}, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

$$R_{\text{MSM}} = 1_A(\text{MSM}) = \begin{cases} 1 & \text{if } d_k^n < \bar{d}^{\text{msm}}, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where  $\bar{t}^{\text{ttc}}$ ,  $\bar{t}^{\text{ttb}}$ , and  $\bar{d}^{\text{msm}}$  are the thresholds of TTC, TTB, and MSM, respectively. They can be modified by international standards for safety [13] or by each system specification. In this paper, the thresholds are set in accordance with the sensor specification of the Co-Pilot system [14] and Korean traffic rules [15].

Through a combination of the risk parameters,  $R_{\text{TTC}}$ ,  $R_{\text{TTB}}$ , and  $R_{\text{MSM}}$ , the primary information of each risk observer results in a single bit, which represents whether its assigned region is safe to enter. The collision risks of the forward and backward risk observers are defined, respectively, as [12]:

$$R_{\text{FORWARD}} = \begin{cases} 1 & \text{if } R_{\text{TTB}} + R_{\text{MSM}} \geq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

$$R_{\text{BACKWARD}} = \begin{cases} 1 & \text{if } R_{\text{TTC}} + R_{\text{MSM}} \geq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

From the definition of each safety factor,  $R_{\text{TTB}}$  and  $R_{\text{MSM}}$  affect the forward risk observers and  $R_{\text{TTC}}$  and  $R_{\text{MSM}}$  affect the backward risk observers.

#### A. Current Forward and Backward Risk Observers

These risk observers report whether the current lane is safe forward and backward. Since we do not consider a reverse maneuver while driving, the result of the backward risk observer does not affect the maneuver decision. The forward risk observer reports the most incoming threat from the nearest vehicle ahead located in the current lane, and its result triggers a lane change maneuver.

#### B. Left Forward and Backward Risk Observers

These risk observers report whether the lane immediately to the left of the vehicle's current lane is safe to enter. To evaluate the collision risks in the adjacent lanes, a virtual vehicle (namely, a ghost vehicle) is used as a substitute of the ego vehicle located at the same curvilinear position in the evaluated lane. Both forward and backward risk observers report the most imminent threat through the nearest vehicle from the ghost vehicle in the left lane. These risk observers check the

safety of the adjacent left lane to change lanes when passing a stopped or slower vehicle ahead, or following a global path that includes a lane change maneuver.

#### C. Right Forward and Backward Risk Observers

These risk observers are the same as the left forward and backward risk observers except that they look to the lane immediately to the right of the vehicle's current lane.

### 2. Velocity Control

This module was designed to determine the desired velocity for collision avoidance when there is a slower vehicle ahead in the driving lane as well as for achieving the target velocity given by the global path. The input data are a relative distance between the ego vehicle and the nearest object, the velocity of the ego vehicle, and the given target velocity. Therefore, the desired velocity is calculated to reduce the following two errors [10]:

$$v_{\text{desired}} = \operatorname{argmin}_{v_{\text{ego}}} \{ \alpha \cdot \varepsilon_1 + \beta \cdot \varepsilon_2 \}, \quad (10)$$

$$\varepsilon_1 = \| v_{\text{ego}} - v_{\text{target}} \|^2 \rightarrow 0, \quad (11)$$

$$\varepsilon_2 = \| d_{\text{nearest}} - f_{\text{rule}}(v_{\text{ego}}) \|^2 \rightarrow 0. \quad (12)$$

To solve (10), we define the cost function as

$$J(v_{\text{ego}}) = \alpha(v_{\text{ego}} - v_{\text{target}})^2 + \beta(d_{\text{nearest}} - f_{\text{rule}}(v_{\text{ego}}))^2. \quad (13)$$

Since (13) is the ordinary least squares model, the gradient descent gives the global minimum as

$$\begin{aligned} \frac{\partial J}{\partial v_{\text{ego}}} &= 2\alpha(v_{\text{ego}} - v_{\text{target}}) + 2\beta(d_{\text{nearest}} - f_{\text{rule}}(v_{\text{ego}})) \left( -\frac{\partial f}{\partial v_{\text{ego}}} \right) \\ &= 2 \left[ \beta d_{\text{nearest}} - \alpha v_{\text{target}} + \alpha v_{\text{ego}} + \beta \left( \frac{\partial f}{\partial v_{\text{ego}}} \right) f_{\text{rule}}(v_{\text{ego}}) \right] \\ &= 0, \end{aligned} \quad (14)$$

where  $v_{\text{ego}}$  represents the velocity of the ego vehicle,  $v_{\text{target}}$  represents the target velocity given by the global path,  $d_{\text{nearest}}$  represents the distance from the nearest object in front, and  $f_{\text{rule}}(v_{\text{ego}})$  represents the rule to follow to achieve the required level of safety;  $\alpha$  and  $\beta$  are weight parameters for each error term.

### 3. Behavior Planning

This module, designed to determine appropriate maneuvers to follow the global path while avoiding collisions, is implemented to perform cooperative driving with a driver as well as automated driving. Based on the concept of identifying a set of driving contexts to execute cooperative driving



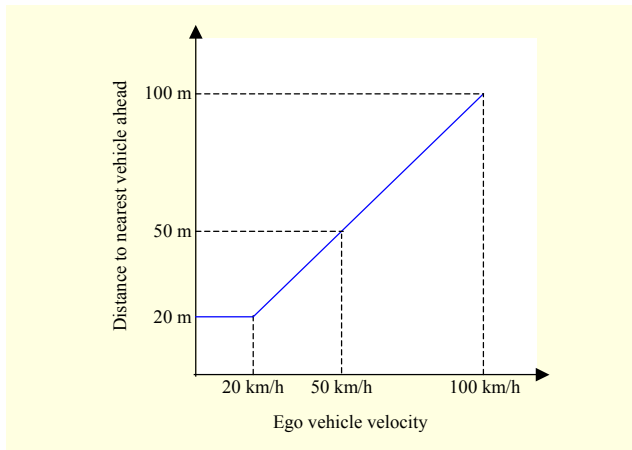


Fig. 6. Function to keep safety distance by traffic rules [16].

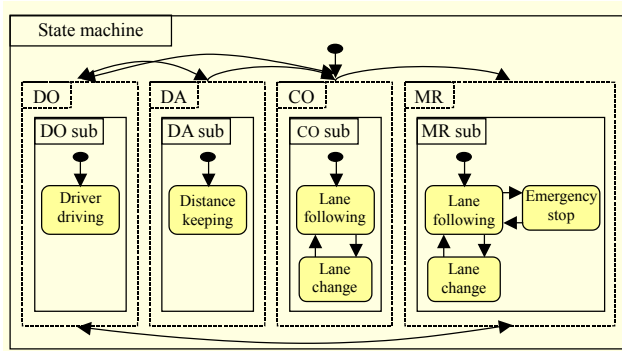


Fig. 7. Diagram of H-FSM. Highest level has four states — DO, DA, CO, and MR. Substates corresponding to each state are driver driving, distance keeping, lane following, lane change, and emergency stop.

between a driver and the system, we use a hierarchical finite state machine (H-FSM) method. At the highest level, we set four states — driver-only (DO) mode, driver-assist (DA) mode, Co-Pilot system (CO) mode, and minimum-risk (MR) mode. The corresponding substates are lane following, lane change, distance keeping, and emergency stop. Figure 7 shows a diagram of the H-FSM with substates corresponding to the states for cooperative driving with a driver and automated driving. Each substate determines which lane to drive in and at what speed.

#### A. DO – Driver Driving

In this substate, the system does not intervene in the driving, and the driver has control authority to change lanes and adjust the vehicle speed by manipulating the steering wheel, throttle, and brake.

#### B. DA – Distance Keeping

This substate, designed to assist the driver in the longitudinal

direction, adjusts the vehicle speed automatically so as not to collide with a slower vehicle ahead, whereas selecting the driving lane is the responsibility of the driver.

#### C. CO – Lane Following

This substate is designed to assist the driver in following the current lane. If there is either a slower or stationary vehicle ahead, then this substate determines not only which lane to drive along based on the results from the left and right risk observers, but also at what speed to drive at. In addition, if there are no adjacent lanes to enter and there is a slower vehicle in front, then the ego vehicle stays in the current lane and adjusts its speed to avoid colliding with the vehicle ahead and maintain a safe distance.

#### D. CO – Lane Change

This substate is designed to assist the driver in changing lanes when the current forward risk observer reports it is unsafe to drive in the current lane. Once this substate has been selected, then the system stays in this state until a lane change has been successfully completed. Given the results of each risk observer for every cycle, this substate determines whether to keep this lane change maneuver or return to the previous lane. For example, the left risk observers report it is safe to enter before the vehicle crosses the lane; however, if it becomes unsafe to enter during this substate and the vehicle still remains in the starting lane, then it determines to stay in the lane.

#### E. MR – Lane Following

This substate, designed to follow the current lane in the MR mode, is the same as lane following in the CO mode except that it only takes the results of the current and right-side risk observers to enter a shoulder safely and stop. In addition, the target velocity is set lower because the MR mode always occurs near a non-confidential region where the system cannot drive for several reasons (poor GPS conditions, inaccurate sensor data, or a different map).

#### F. MR – Lane Change

This substate, designed for changing lanes, is the same as lane change in the CO mode except that it only takes the results of the current and right-side risk observers. Furthermore, it checks whether a shoulder exists. If there is no shoulder, then it executes an emergency stop maneuver.

#### G. MR – Emergency Stop

This substate, designed to stop in the case of an emergency, maintains the current lane and executes a “zero velocity” directive.

## V. Trajectory Generation

This section details the trajectory generation method to follow or change lanes by generating trajectories of a vehicle. The trajectory generation module selects four control points on the current trajectory and desired lane centerline model, and then connects the selected control points using G2CBS smoothing. Using this method, continuous trajectories can be generated to follow or change lanes without abrupt heading changes.

### 1. Waypoint Selection

The waypoint selection is one step used to generate the trajectories of a vehicle by selecting four different waypoints. Figure 8 shows the concept of the waypoint selection in the trajectory generation method. The first waypoint is set as the current position of the vehicle. The second waypoint is selected from the current trajectory using the following equation [17]:

$$w_1 = \underset{p_i}{\operatorname{argmin}} \left( |w_0 - p_i| - \alpha L_{\text{traj}} \right), \quad (15)$$

$$\left( p_i \in P_c, \frac{p_i - w_0}{|p_i - w_0|} \cdot \delta_c \leq G_{\max} \right),$$

where  $w_0$  is the first waypoint corresponding to the current position of the vehicle;  $p_i$  is a node on the regularly discretized current trajectory  $P_c$ ;  $\alpha$  is a coefficient;  $L_{\text{traj}}$  is the length of the trajectory determining the current velocity of the vehicle and reference velocity of the road;  $\delta_c$  is the unit vector of the heading angle of the vehicle; and  $G_{\max}$  is the threshold value that represents the steering limit of the vehicle. The third and fourth waypoints are selected from the desired lane and are described as follows [17]:

$$w_2 = \underset{q_i}{\operatorname{argmin}} \left( |w_0 - q_i| - (1 - \alpha) L_{\text{traj}} \right),$$

$$w_3 = \underset{q_i}{\operatorname{argmin}} \left( |w_0 - q_i| - L_{\text{traj}} \right), \quad (16)$$

$$\left( q_j \in Q_d, \frac{q_j - w_0}{|q_j - w_0|} \cdot \delta_c \leq G_{\max} \right),$$

where  $q_j$  is a node on the regularly discretized desired lane model  $Q_d$ . If the decision-making module sets the current lane occupied by the vehicle as the desired lane to follow, then the third and fourth waypoints are selected from the current lane model. In addition, if the decision-making module sets the adjacent lanes as the desired lane to pass a slower or a stopped vehicle ahead, then the third and fourth waypoints are selected from the adjacent lanes.

### 2. G2CBS Smoothing

The selected waypoints are connected by G2CBS smoothing,

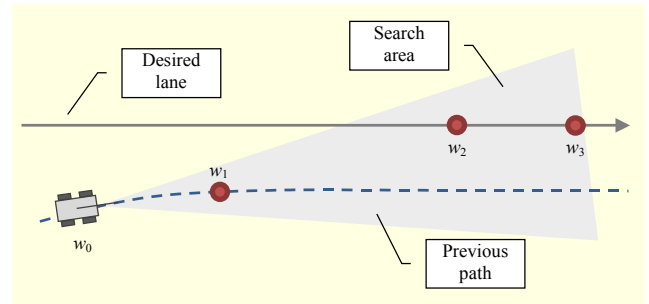


Fig. 8. Waypoint selection: proposed method selects four waypoints from current vehicle position, current trajectory, and desired lane. Blue dashed line, solid black line, and red dots represent current trajectory, desired lane, and selected waypoints, respectively. Gray area represents search area considering steering limit of vehicle [17].

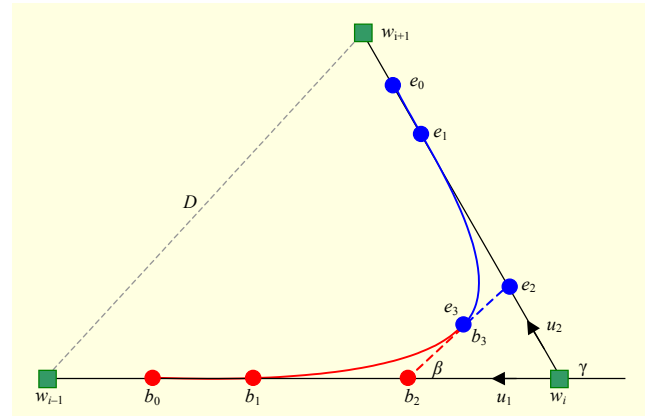


Fig. 9. Two cubic Bézier spiral curves of G2CBS smoothing to interpolate three waypoints. Green rectangles represent three waypoints, red and blue curves represent first and second cubic Bézier spiral curves, respectively, and red and blue dots represent control points of each corresponding curve [17].

which is an algorithm used to interpolate three adjacent points on a 2-D plane by generating a  $G^2$  continuous curve that satisfies  $G^1$  (tangent) and  $G^2$  (curvature) continuities. The G2CBS smoothing algorithm is composed of two cubic Bézier spiral curves that satisfy the following conditions [17]–[18]:

$$F(\beta) = \cos^2(\gamma - \beta) \sin \beta$$

$$\times \left[ (c_1 + 6 \cos^2 \beta) D_y - 6 D_x \cos \beta \sin \beta \right]$$

$$+ \cos^2 \beta \sin(\gamma - \beta)$$

$$\times \left[ c_1 (D_y \cos \gamma - D_x \sin \gamma) + 6 \cos(\gamma - \beta) \right]$$

$$\times (D_y \cos \beta - D_x \sin \beta)$$

$$= 0, \quad (17)$$

where  $\gamma$  is the angle between the two lines  $\overline{w_i w_{i-1}}$  and  $\overline{w_i w_{i+1}}$ ;  $c_1$  is the coefficient for the cubic Bézier spiral

segments; and  $\mathbf{D} = (D_x, D_y)$  is the distance between  $w_{i-1}$  and  $w_{i+1}$ . If  $\beta$  is defined, then eight control points are determined to construct two cubic Bézier spiral curves, as shown in Fig. 9. The eight control points of the two cubic Bézier spiral curves are defined as follows [17]–[18]:

$$\begin{aligned} b_0 &= w_i + d_1 \mathbf{u}_1, & b_1 &= b_0 - g_b \mathbf{u}_1, \\ b_2 &= b_1 - h_b \mathbf{u}_1, & b_3 &= b_2 + k_b \mathbf{u}_d, \\ e_0 &= w_i + d_2 \mathbf{u}_2, & e_1 &= e_0 - g_e e_2, \\ e_2 &= e_1 - h_e \mathbf{u}_2, & e_3 &= e_2 - k_e \mathbf{u}_d, \end{aligned} \quad (18)$$

where  $b_0, b_1, b_2,$  and  $b_3$  are four control points of the first cubic Bézier spiral curve, and  $e_0, e_1, e_2,$  and  $e_3$  are four control points of the second cubic Bézier spiral curve;  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are unit vectors along lines  $w_i w_{i-1}$  and  $w_i w_{i+1}$ , respectively;  $\mathbf{u}_e$  is unit vector along lines  $b_2 e_2$  and other variables are defined as follows [17]–[18]:

$$\begin{aligned} h_b &= \frac{(c_2 + 4)D_y \cos^2(\gamma - \beta) \sin \beta}{[(c_1 - 3)\cos \beta \sin(\gamma - \beta) + 6\sin \gamma] \times \cos \beta \sin \gamma}, \\ h_e &= \frac{h_b \cos^2 \beta \sin(\gamma - \beta)}{\sin \beta \cos^2(\gamma - \beta)}, \\ g_b &= c_2 h_b, & g_e &= c_2 h_e, \\ k_b &= \frac{6}{c_2 + 4} h_b \cos \beta, \\ k_e &= \frac{6}{c_2 + 4} h_e \cos(\gamma - \beta), \\ c_2 &= \frac{2}{5}(\sqrt{6} - 1). \end{aligned} \quad (19)$$

Unfortunately,  $\gamma$  is the only observable variable, and  $\beta$  can be determined using a numerical method when  $d_1$  and  $d_2$  are given. To find  $\beta$  analytically without a numerical method, the following conditions are to be met [17]–[18]:

$$\begin{aligned} d &= d_1 = d_2, \\ \beta &= \frac{\gamma}{2}. \end{aligned} \quad (20)$$

From these conditions, the variables used to determine eight control points of two cubic Bézier spiral curves are easily acquired as follows [17]–[18]:

$$\begin{aligned} h_b &= h_e = c_3, \\ g_b &= g_e = c_2 c_3 d, \\ k_b &= k_e = \frac{6c_3 \cos \beta}{c_2 + 4} d, \\ c_3 &= \frac{c_2 + 4}{c_1 + 6}. \end{aligned} \quad (21)$$

## VI. Experimental Results

The system was tested on a closed road with other vehicles

to verify whether it works properly for cooperative driving with a driver and automated driving by determining the appropriate maneuvers and generating trajectories to deal with various situations that are similar to real road environments. The in-vehicle test was closely intertwined with an algorithm development through a simulation procedure with logged data replies in ROS, on-site testing for debugging, and a system evaluation. Figure 10 shows a flow diagram for a Co-Pilot agent in ROS. The experiments were conducted using two scenarios. The first scenario was conducted to verify that the system operates properly for cooperative driving with a driver and a minimum risk mode when a driver denies or cannot take over the control when the Co-Pilot system requests it. The second scenario was conducted to verify that the system runs in complicated driving situations while avoiding collisions; for example, to pass a slower vehicle ahead. The graphs shown in Figs. 11–14 illustrate the integrated test results of the system for the scenarios. The detailed test result of the integrated in-vehicle tests is available at <http://youtu.be/Pa7nm4IzcNk> and <http://youtu.be/rgDvJki9yj0>.

In these graphs, the magenta, green, white, and blue lines represent the current lane, adjacent lanes available to drive in, other lanes unavailable to drive in, and the generated trajectory, respectively. The white car image represents the ego vehicle. The magenta and green rectangles represent the tracked objects projected onto the map. White messages on the right side represent the current maneuver information.

### 1. Scenario I

The first scenario consists of several services such as driver driving, distance keeping, lane following, lane changes, minimum risk of stopping, and driving mode transitions. The details of the scenario, as shown in Fig. 13, are as follows: (a) a driver drives the vehicle in DO, (b) the vehicle drives in DA when the driver pushes the ACC button, (c) the vehicle drives in CO when the Co-Pilot agent detects that the driver is getting drowsy, and (d) the drive mode is changed to MR when the vehicle approaches a non-confident region, and the vehicle then stops safely in the shoulder.

In DA, as shown in Fig. 13(c), the system drove the vehicle while keeping an appropriate safe distance from a slower vehicle ahead without collisions by controlling the speed of the vehicle. If the distance from the slower vehicle ahead is shorter than the safe distance, then the system reduces the speed of the vehicle, as shown Fig. 11. The safe distance from the vehicle ahead is determined by considering the current speed of the vehicle and the target velocity of the road.

In CO, as shown in Figs. 13(d)–13(f), the system takes control of the vehicle. The system determines the vehicle



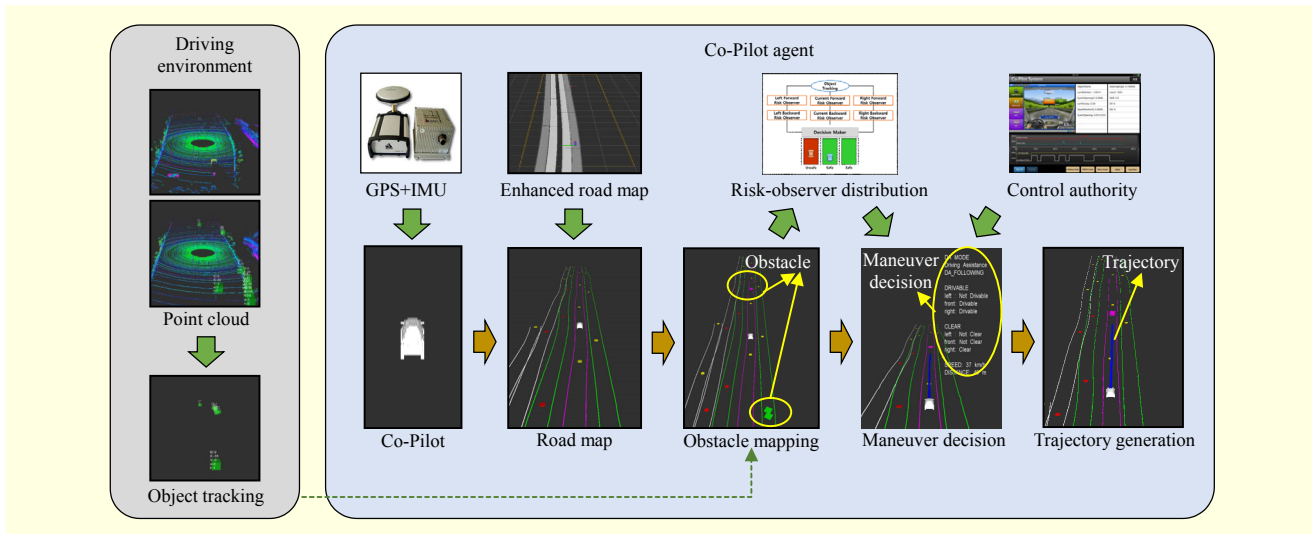


Fig. 10. Flow diagram of Co-Pilot agent in ROS. Co-Pilot agent consists of four nodes — road map that builds and publishes road map data; risk-observer distribution that recognizes and evaluates current traffic situations; decision-making that determines appropriate maneuvers based on results of risk observers and control authorities; and trajectory generation that generates trajectories to achieve given maneuvers.

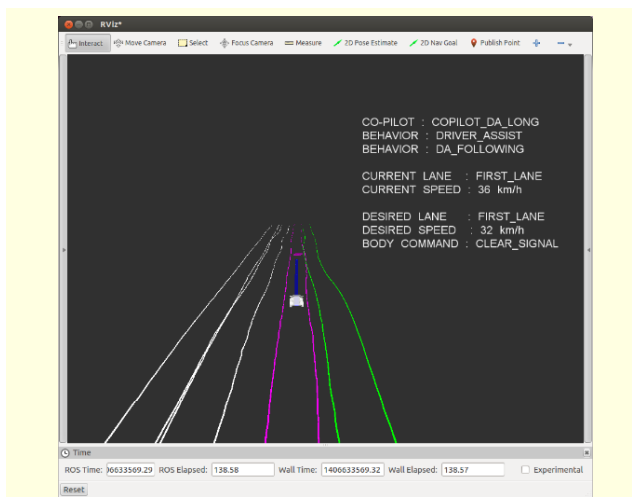


Fig. 11. Detailed figure showing experimental result of scenario I (DA). System adjusts speed of ego vehicle to maintain appropriate safe distance from slower vehicle ahead.

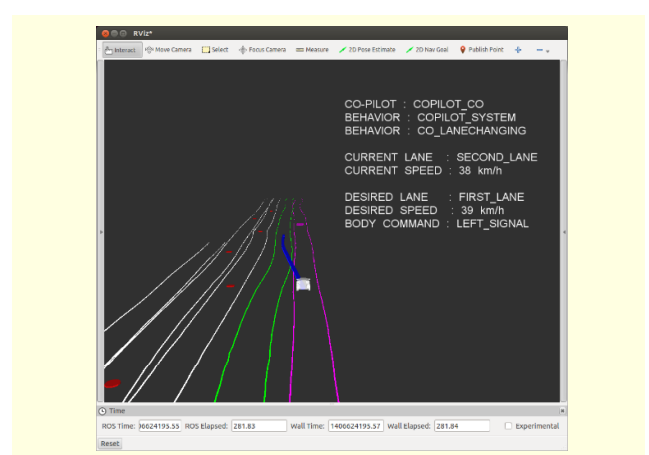


Fig. 12. Experimental results of scenario I (CO). System determines lane change maneuver and generates lane change trajectory to pass slower vehicle ahead.

maneuvers by integrating various types of information, such as a map, positions of the ego vehicle, and other tracked vehicles. When the current forward risk observer reports the current lane as unsafe because of a slower or stopped vehicle ahead, while the left risk observers report it to be safe and the right risk observers report it to be non-drivable, the system chooses to take the adjacent left lane and passes the slower vehicle ahead, as shown in Fig. 12.

The system changes the drive mode from DO to CO by detecting the driver's status. When the system detects that the driver is becoming drowsy, the system warns the driver through multimodal interfaces, such as a warning sound or

messages, and asks the driver's agreement to take control of the vehicle. If the driver agrees, then the system takes control of the vehicle and drives autonomously.

In MR, as shown in Figs. 13(g)–13(i), the system conducts a minimum risk maneuver to guarantee the safety of the vehicle. The driving mode is transferred to the MR mode when the vehicle approaches a non-confidential region. In this situation, the system recommends that the driver takes over control of the vehicle. If the driver takes control of the vehicle, then the drive mode is changed to DO. Otherwise, the system changes the drive mode to MR because the driver rejects the recommendation or does not recognize it.

The experiment was successfully conducted and verified that

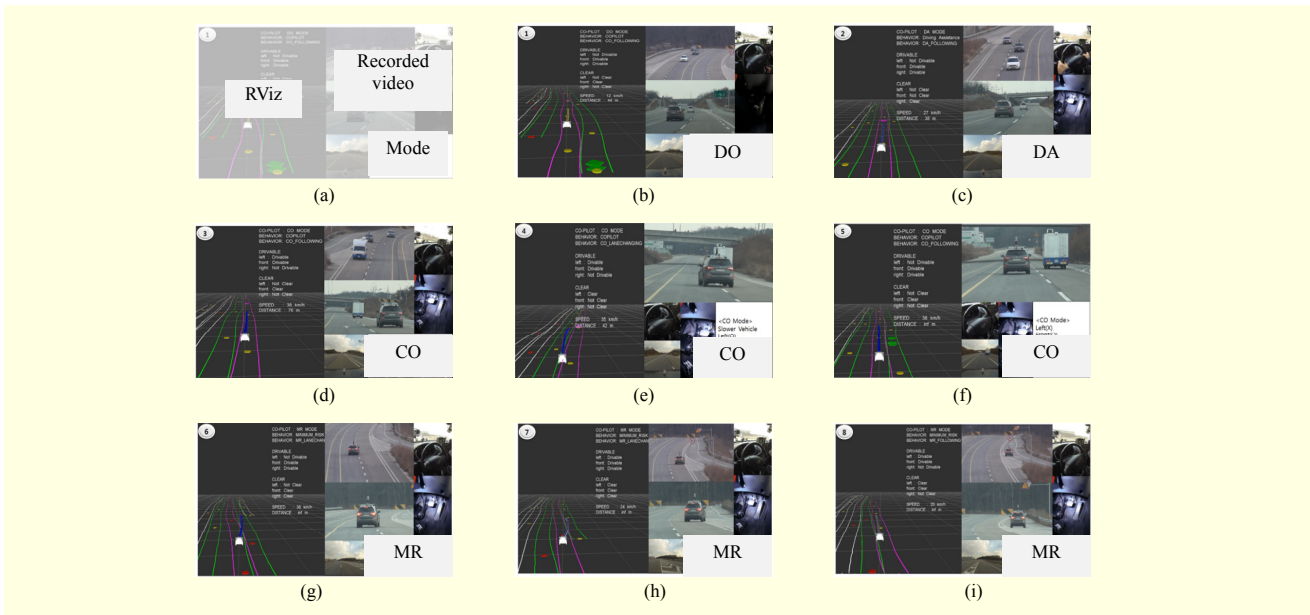


Fig. 13. Experimental results of scenario I: (a) message layout, (b) DO mode, (c) DA mode, (d–f) CO mode, and (g–i) MR mode.

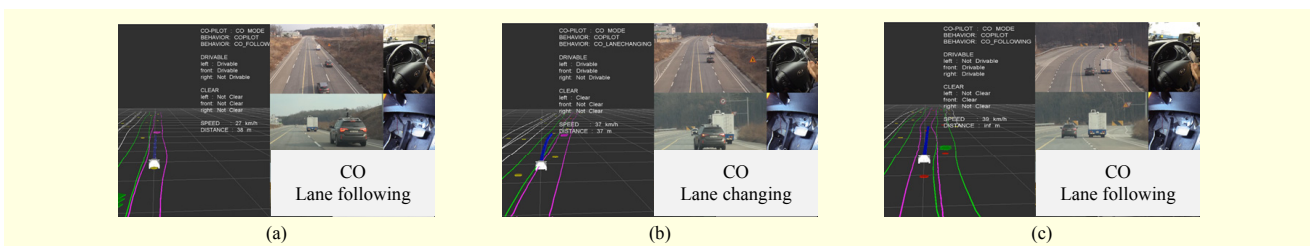


Fig. 14. Experimental result of scenario II: (a) vehicle starts in right lane and follows slower vehicle ahead by maintaining safe distance, (b) system starts to take left lane to pass slower vehicle ahead, and (c) vehicle follows left lane at maximum speed of road.

our system can provide several services for cooperative driving with a driver and autonomous driving.

## 2. Scenario II

The second scenario was conducted under a complicated situation, such as passing a slower vehicle ahead while being surrounded with other vehicles. The details of the scenario are as follows: (a) the vehicle starts in the right lane and follows a slower vehicle ahead by maintaining a safe distance until the adjacent left lane is safe to enter, as shown in Fig. 14(a), (b) if the left lane is clear, then the system generates the trajectory to pass the slower vehicle ahead by determining the lane change maneuver to move to the adjacent left lane, as shown in Fig. 14(b), and (c) the vehicle follows the left lane with the target speed of the road if there is no slower vehicle ahead, as shown in Fig. 14(c).

The experiment was successfully conducted and verified that the system can determine maneuvers and generate trajectories appropriately in a complicated situation without collisions.

## VII. Conclusion

In this paper, we proposed a system architecture and algorithms for cooperative driving between the driver and the Co-Pilot system. The Co-Pilot system can determine the behaviors and generate the trajectories for various driving situations. To verify the functional correctness and measure the performances, the system was tested on a closed road with other moving vehicles. As the experimental results showed, the system operated properly in several scenarios that are similar to a real road environment. We expect the Co-Pilot system to be a platform that can be used in developing Advanced Driver Assistance System applications as well as autonomous driving systems. Future works are planned to verify our system on a real road if legal issues are resolved, and to consider human factors to adapt the system parameters for comfortable driving.

## References

- [1] J.D. Rupp and A.G. King, "Autonomous Driving - A Practical

- Roadmap,” SAE Technical Paper 2010-01-2335, 2010.
- [2] K. Aldana, “U.S. Department of Transportation Releases Policy on Automated Vehicle Development,” National Highway Traffic Safety Administration (NHTSA) New Release, May 2013.
- [3] SAE International, “Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems,” Technical Report SAE J 3016, Jan. 2014.
- [4] C. Urmson et al., “Autonomous Driving in Urban Environments: Boss and the Urban Challenge,” *J. Field Robot.*, vol. 25, no. 8, Aug. 2008, pp. 425–466.
- [5] A. Bacha et al., “Odin: Team VictorTango’s Entry in the DARPA Urban Challenge,” *J. Field Robot.*, vol. 25, no. 8, Aug. 2008, pp. 467–492.
- [6] M. Montemerlo et al., “Junior: The Stanford Entry in the Urban Challenge,” *J. Field Robot.*, vol. 25, no. 9, Aug. 2008, pp. 569–597.
- [7] ROS. [Online], <http://www.ros.org/>
- [8] S.Y. Cho and H.K. Lee, “Modified RHKF Filter for Improved DR/GPS Navigation against Uncertain Model Dynamics,” *ETRI J.*, vol. 34, no. 3, June 2012, pp. 379–387.
- [9] M. Enkhtur, S.Y. Cho, and K.-H. Kim, “Modified Unscented Kalman Filter for a Multirate INS/GPS Integrated Navigation System,” *ETRI J.*, vol. 35, no. 5, Oct. 2013, pp. 943–946.
- [10] S. Glaser et al., “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, Sept. 2010, pp. 589–606.
- [11] B. Patrick et al., “Multi-agent Interactions in Urban Driving,” *J. Physical Agents*, vol. 2, no. 1, Mar. 2008, pp. 15–30.
- [12] S. Noh, K. An, and W. Han, “Situation Assessment and Behavior Decision for Vehicle/Driver Cooperative Driving in Highway Environments,” in proc. of *IEEE Int. Conf. Autom. Sci. Eng.*, Gothenberg, Sweden, Aug. 24–28, 2015, pp. 626–633.
- [13] ISO 22179, *Intell. Transp. Syst. – Full Speed Range Adaptive Cruise Contr. (FSRA) Syst. – Performance Requirements and Test Procedures*, 2009.
- [14] S. Noh, K. An, and W. Han, “High-Level Data Fusion Based Probabilistic Situation Assessment for Highly Automated Driving,” *IEEE Int. Conf. Intell. Transp. Syst.*, Las Palmas de Gran Canaria, Spain, Sept. 15–18, 2015, pp. 1587–1594.
- [15] Korean Traffic Laws. [Online], <http://www.law.go.kr/>
- [16] S. Noh and W. Han, “Independent Object Based Situation Awareness for Autonomous Driving in On-Road Environment,” *J. Inst. Contr., Robot. Syst.*, vol. 21, no. 2, Feb. 2015, pp. 87–94.
- [17] B. Park, Y.-C. Lee, and W. Han, “Trajectory Generation Method Using Bezier Spiral Curves for High-Speed On-Road Autonomous Vehicles,” in proc. of *IEEE Int. Conf. Autom. Sci. Eng.*, Taipei, Taiwan, Aug. 18–22, 2014, pp. 927–932.
- [18] K. Yang and S. Sukkarieh, “Anytime Synchronized-Biased-Greedy Rapidly-Exploring Random Tree Path Planning in Two

Dimensional Complex Environments,” *Int. J. Contr., Autom., Syst.*, vol. 9, no. 4, Aug. 2011, pp. 750–758.



vehicles and robots.

**Samyeul Noh** received his BS degree in systems science from the School of Engineering Science, Osaka University, Japan, in 2009 and his MS degree in electrical engineering from the University of San Diego, USA, in 2012. Since 2012, he has been with ETRI. His current research interests are navigation for autonomous



autonomous vehicles.

**Byungjae Park** received his BS, MS, and PhD degrees in mechanical engineering from Pohang University of Science and Technology, Rep. of Korea, in 2005, 2007, and 2013, respectively. Since 2013, he has been with ETRI. His current research interests are mainly concentrated on the navigation and mapping of mobile robots and



main research interests include map building and provisioning platforms for autonomous vehicles; navigation services; and location-based services.

**Kyoungwan An** received his BS, MS, and PhD degrees in computer science from Pusan National University, Rep. of Korea, in 1997, 1999, and 2004, respectively. Since 2004, he has been with ETRI. He is currently serving as principal researcher for the IT Convergence Technology Research Laboratory at ETRI. His



His research interests include distributed systems, machine learning, autonomous driving systems, and human-machine interaction.

**Yongbon Koo** received his BS and MS degrees in computer science from the School of Computing, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2004 and 2006, respectively. Since 2006, he has been with ETRI. He is currently pursuing his PhD degree in computer



**Wooyong Han** received his BS and MS degrees in electronics engineering from Kyung Hee University, Yongin, Rep. of Korea, in 1983 and 1985, respectively. He received his PhD degree in computer science from Chungnam National University, Daejeon, Rep. of Korea, in 2005. Since 1989, he has been with ETRI,

where he is now a principal researcher. His main research interests are object-oriented distributed computing; XML-based e-commerce common frameworks; XML-based B2B system interoperability; XML digital signature and encryption; and frameworks for telematics applications independent of mobile networks, vehicles, and territorial digital multimedia broadcasting.