

자율주행 셔틀버스의 통신 정보 융합 기반 충돌 위험 판단 알고리즘 개발

이승민* · 이창형** · 박만복***

Development of I2V Communication-based Collision Risk Decision Algorithm for Autonomous Shuttle Bus

Seungmin Lee*, Changhyung Lee**, Manbok Park***

Key Words : Autonomous driving(자율주행), V2X-communication(차량간 통신), Collision estimation(충돌 예측), Autonomous emergency brake(자동긴급 제동), Cooperative Sensing(협력 센싱)

ABSTRACT

Recently, autonomous vehicles have been studied actively. Autonomous vehicles can detect objects around them using their on board sensors, estimate collision probability and maneuver to avoid colliding with objects. Many algorithms are suggested to prevent collision avoidance. However there are limitations of complex and diverse environments because algorithm uses only the information of attached environmental sensors and mainly depends on TTC (time-to-Collision) parameter. In this paper, autonomous driving algorithm using I2V communication-based cooperative sensing information is developed to cope with complex and diverse environments through sensor fusion of objects information from infrastructure camera and object information from equipped sensors. The cooperative sensing based autonomous driving algorithm is implemented in autonomous shuttle bus and the proposed algorithm proved to be able to improve the autonomous navigation technology effectively.

1. 서론

운전자의 개입 없이 주변 환경을 인식하여 스스로 움직이는 자율주행 자동차의 연구는 많은 기관에서 폭넓게 연구되고 있다. 최근 여러 기업과 학교에서는 자율주행 자동차의 상용화를 위한 개발 및 시연이 이뤄지고 있다.^(1,2) 이전 연구들에서는 자기 차량 센서와 환경 센서를 적용하여 객체 및 주변 환경을 인식하고 판단하여 제어를 수행하고 있으나, 환경 센서들은 인식 범위의 한계로 인하여

인식 거리 범위 밖의 객체, 센서 FOV(Field of View)를 벗어나는 사각지대의 객체, 또는 센서 Occulusion 등으로 인해 객체들을 인식할 수 없는 경우가 많다.^(3,4) 이러한 이유로 인해 실제 시연에서는 제한된 지역 및 제한된 기능으로 시연을 하는 수준이다.

본 연구에서는 국내 최초로 인프라 정보 융합 기반으로 차량과 관제와의 통신을 통해 교내를 주행하는 순환식 셔틀버스를 개발하였으며 환경 센서들을 통한 주변 물체의 검출뿐만 아니라 사각지대에 대한 물체의 검출을 통하여 차량과 보행자 등이 혼재된 혼잡 상황 하에서도 자율주행이 가능한 협력 센싱 자율주행 시스템을 개발하였다. 협력 센싱 자율주행 시스템은 자율주행셔틀버스, 무선 통신 및 카메라 인프라, 관제 센터 시스템으로 구성되어 있

* 한국교통대학교 일반대학원 전자공학과 & Garage.M

** 한국교통대학교 글로벌대학원 전자공학과 & 아이우도반

*** 한국교통대학교 전자공학과

E-mail : movingsm@naver.com

다. 자율주행 셔틀버스의 시스템 구성으로는 라이다(Lidar), 카메라(Camera), IMU, GPS등이 있다. 자율주행 버스는 다른 자율주행 자동차와 마찬가지로 GPS를 통한 위치 추적뿐만 아니라 교내 도로의 정밀지도 정보를 수집하여 예측경로를 생성하며, 차량에 부착된 라이다(Lidar)를 통하여 도로상에 존재하는 물체들의 검출을 수행하였다. 또한 차량에 부착된 통신용 단말기를 통하여 인프라 카메라로부터 획득한 객체 정보를 획득하게 된다. 무선 통신 및 카메라 인프라에서는 카메라를 통해 획득한 정보를 관제 센터 및 차량에 전달하게 된다. 관제 센터 시스템에서는 차량운행 관리 및 전체 시스템 모니터링 등이 이루어지게 된다.

본 연구에서 기존 연구에서 개발 적용되었던 TTC(Time to Collision) 알고리즘과 긴급제동 알고리즘 뿐만 아니라 협력 센싱 정보를 이용한 충돌 알고리즘을 추가하여 차량과 보행자 등이 혼재된 혼잡류 상황 하에서도 자율주행 4단계가 가능한 자율주행 자동차의 충돌 예측 알고리즘을 개발하였으며 이를 실제 교내 셔틀 버스 경로 상에서 구현하고 검증하였다.

자율주행자동차의 개발에 한양대학교에서는 레이저 스캐너를 통하여 전방에 대한 물체의 충돌을 예측하는 알고리즘을 제시하였으며,⁽⁵⁾ 레이저 스캐너가 감지할 수 있는 구역에 대해서 충돌 판단 시스템을 제안하여 자기 차량과 충돌 물체간의 충돌 시간과 충돌 읍셋량을 계산하기 위한 궤적 예측 방법을 제시하였다. 국민대 또한 Lidar 센서를 통하여 포인트 클라우드 데이터들의 군집화와 분류를 통하여 Lidar가 검출할 수 있는 주변 물체들의 추출을 제시하였다.^(6,7) 차량간 통신을 이용하여 충돌 확률을 예측한 연구도 있다.⁽⁸⁾ 레이더 기반 인지 성능에 대한 정량적 평가 연구도 있다.⁽⁹⁾ 또한 차선 유실구간 측위를 위한 레이저 스캐너 기반 고정 장애물 탐지 알고리즘 개발 사례도 있다.⁽¹⁰⁾

다만 위와 같은 경우는 전방의 물체를 Lidar와 같은 센서로만 감지하며, 실제로 사람도 확인하기 힘든 사각지대에 대한 경우에는 Lidar와 같은 센서로도 물체를 빠르게 감지할 수가 없어서 사고의 위험이 생길 수 있다. 본 논문에서는 사각지대에 대한 사고의 위험을 줄이기 위하여 추가적으로 사각지대에 존재하는 물체에 대한 정보를 Infrastructure camera로 받아서 실제 운행 중 생기는 충돌 위험 순간에 대한 실차 정보를 받아 실험하였다.

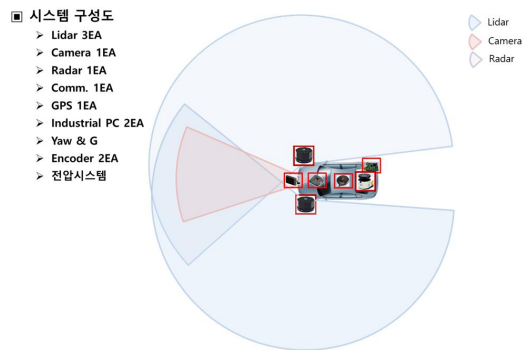
2. 시스템 구성

2.1. 시험 차량 구성

시험에 사용된 차량은 Fig. 1의 차량으로, 국립한국교통대학교에서 자체적으로 제작한 차량인 아이브(IVE, 이후 Subject 차량으로 정의)를 사용하였다. 차량의 프레임 제작은 대창모터스社에서 제작을 하였으며, 차량의 전방에는 Quanergy社에서 만든 8개의 채널을 가진 Lidar를 양 측으로 1대씩 총 2대를 장착하였다. Lidar의 경우 sampling rate를 100msec로 설정하였다. 전방에는 초음파 센서를 설치하여 근접한 충돌 사고를 방지하고자 하였다. 후미에는 Sick社에서 만든 레이저 스캐너 1대를 설치하여 후방 접근 객체들에 대한 인식을 수행하였으며 sampling rate를 100msec로 설정하였다. 차량의 위에는 GPS 장치와 Infrastructure camera와의 통신 장치를 위한 안테나가 설치되었다. GPS장치는 상용 제품으로 1Hz의 주기를 가지며 차량간 통신 장치의 경우 100msec마다 송수신하는 것으로 설정하였다. 차량의 아래에는 자기



(a) Picture of Test Vehicle



(b) System Configuration

Fig. 1 Test Vehicle

장의 세기를 감지하여 차량의 위치를 판단하는 자기장 센서가 설치되어 있다. 차량의 내부에는 데이터 처리를 위한 산업용 컴퓨터가 설치가 되어있으며, Lidar 데이터와 통신용 장치의 데이터는 이더넷을 통하여 산업용 컴퓨터와 통신을 하며, GPS데이터를 포함하여 USB 단자를 통한 CAN 통신과 시리얼 통신을 사용하여 산업용 컴퓨터로 모든 데이터를 수집하며 통신시간을 포함하여 지연시간, 영상의 압축 및 처리 시간 등 전 처리 과정에 대한 Timer를 설정하여 시간동기화를 실시한다.

Subject 차량은 차량이 움직이는 동안 상대좌표로 정의가 되어 차량의 전방 차축의 중앙을 기준으로 차량은 좌표상의 기준점 (0,0)로 정의한다. 기준점을 기준으로 차량의 폭은 1.6m, 길이는 4m가 된다. 차량은 좌표를 기준으로 X축 방향으로 이동하며, 차량이 좌회전 시에는 Y축의 양수부분으로, 우회전 시에는 Y축의 음수방향으로 이동을 하게 된다.

Subject 차량의 속도는 최대 4m/s이며 차량에서 사용하는 단위는 길이(m), 속도(m/s)로 정의한다. Subject 차량은 Fig. 2에 나오는 교내의 코스를 순환하며 중간에 설치된 버스정류장에서 정차를 한다.



Fig. 2 Course for Shuttle Bus

2.2. 알고리즘 아키텍처

Subject 차량의 전체적인 알고리즘은 Fig. 3과 같다.

Path Generation(SW4)에서 차량은 GPS를 사용하여 차량의 위치를 확인한 뒤 정밀 지도를 통하여 차량의 예측 경로를 직선 도로 혹은 회차로에 대하여 차량이 운행 가능한 범위 내의 구간에 대한 유사한 이차방정식($ax^2 + bx$)으로 생성하여, 직선도로의 경우 이차항의계수가 0에 수렴하게 되는 예측경로를 생성하여, Object Processing(SW1, SW2)에서는 차량에 설치된 각각의 Lidar 센서를 사

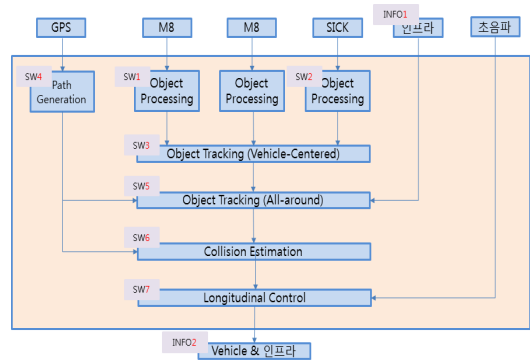


Fig. 3 SW Architecture for Autonomous Bus

용하여 차량 주변에 존재하는 물체들의 Rawdata(x,y)를 검출하며, Object Tracking(SW3, SW5)에서는 Rawdata(x,y)로 이뤄진 물체들에 대하여 군집화 처리를 하여 검출한 물체의 중심점 object center(x,y), 그 물체에 대하여 X축으로 가장 가까운 위치에 존재하는 edge인 object minx(x,y), Y축으로 가장 가까운 위치에 존재하는 object miny(x,y)를 생성하여 Object의 크기에 대해서도 보상을 하여 더욱 강건한 알고리즘을 수행하게 도와주고, 추가적으로 Infra-structure Camera에서 검출하는 사각지대의 물체들도 object camera(x,y)로 검출을 한다. Collision Estimation(SW6)에서는 SW4와 SW5에의 데이터들을 이용하여 차량이 운행하는 동안 예측 경로에 주변에 존재하는 Object들의 속도를 계산해 충돌의 가능성이 있는 물체에 대한 검출과 함께 warning flag를 생성하며, Longitudinal Control(SW7)에서는 warning flag에 따라 실제로 차량을 출발과 멈춤을 제어한다. 위 모든 과정 처리는 한 제어주기 당 20ms로 처리되며, Fig. 4와 같은 화면을 통해 모니터링을

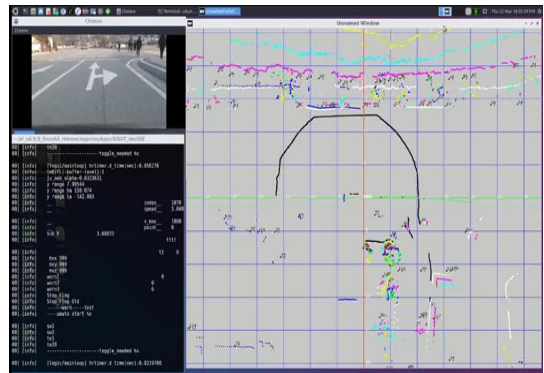


Fig. 4 Test Monitoring System

Table 1 Test Vehicle

Data	Definition
$Path_{sub} = ax^2 + bx$	The predicted path of the Subject
V_{sub}	The speed of the Subject (m/s)
$(x, y)_{obj}$	Object
$(x, y)_{cent}$	The center of the Object
$(x, y)_{minx}$	The minimum x of the Object
$(x, y)_{miny}$	The minimum y of the Object
$(Vx, Vy)_{cent}$	Direction of the cent
$(Vx, Vy)_{minx}$	Direction of the minx
$(Vx, Vy)_{miny}$	Direction of the miny

실시한다. 본 연구에서는 Collision Estimation(SW6)에 대한 3가지 알고리즘을 소개하며, 변수의 정의는 Table 1과 같다.

3. 충돌 위험 판단 알고리즘 개발

3.1. 충돌 위험 판단 알고리즘 아키텍처 구조

Subject 차량의 충돌 예측 알고리즘(Collision Estimation, SW5)은 크게 3가지를 통하여 충돌을 예측하며, 3가지 상황 중 하나라도 발생한다면 충돌 안전을 위해 일단 차량 정지를 수행하여 최소한 안전을 확보한다. $Path_{sub}$ 에서 Object 물체가 충돌을 할 상황이 생길 경우, 충돌까지 소요되는 시간을 측정하여 충돌 예측을 확인하는 Time to Collision Algorithm(TTC)과, 긴급한 상황에서 차량의 제어를 강제적으로 제동하는 긴급 제동 Algorithm, 마지막으로 Lidar로 측정할 수 없는 사각지대에서 오는 물체에 대하여 Infrastructure camera로 측정된 협력 센싱 기반의 충돌 예측 Algorithm이 있다.

3.2. 충돌 위험 판단 알고리즘

3.2.1. TTC 기반 충돌 예측 알고리즘 개발

다음 Fig. 5는 TTC 기반 알고리즘에 대한 아키텍처를 보여주고 있다.

위 알고리즘에서는 Fig. 5와 같이 Object의 위치와 방향을 사용하여 Object의 예측 경로 $Path_{obj}$ 를 생성하여, $Path_{obj}$ 와 $Path_{sub}$ 사이에서 생기는 교점을 충돌예측지점

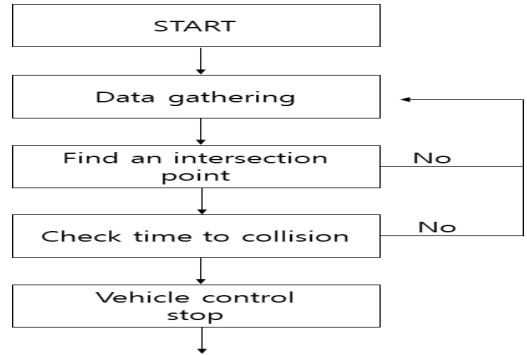


Fig. 5 Architecture for TTC Algorithm

$Cp(x, y)$ 를 정하여 Subject와 Object가 $Cp(x, y)$ 까지 도달하는데 걸리는 시간으로 충돌을 예측을 하며, Fig. 6과 같이 Subject가 주행 중인 상황에서 예측경로 상에 Object의 예측 경로와 충돌이 일어나는 점을 찾아내며, 현재 이동 경로 $Path_{obj}$ 는 식 (1)로 정의한다.

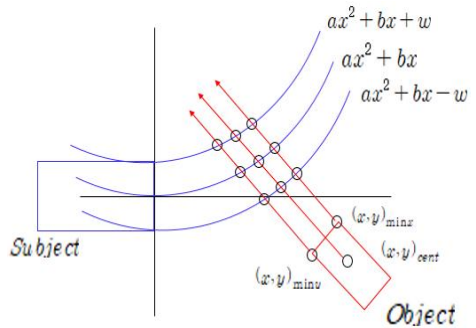


Fig. 6 Concept of TTC Algorithm

$$Path_{obj} = \frac{Vy_{obj}}{Vx_{obj}}(x - x_{obj}) + y_{obj}, \quad (1)$$

$$\begin{aligned} \text{단 } x_{obj} &= x_{cent.minx.miny}, \quad y_{obj} = y_{cent.minx.miny}, \\ Vx_{obj} &= Vx_{cent.minx.miny}, \quad Vy_{obj} = Vy_{cent.minx.miny} \end{aligned}$$

단, 식 (1)에서 Vx 가 0이 되는 경우에는 Subject와 Object가 수직인 상황으로 만나는 상황으로, 이 경우 일차항의 기울기 Vy_{obj}/Vx_{obj} 는 0으로 설정하도록 하였다. 하나의 object에 대한 중점뿐만 아니라 minx와 miny에 대한 예측 경로도 생성하여 Object마다 3개의 $Path_{obj}$ 가 생겨서 크기에 대한 보상을 해줄 수 있게 되며, $Path_{obj}$ 를 좌변에,

$Path_{sub}$ 을 우변에 둔 뒤 둘 사이에 공통의 교점 (x, y) 을 식 (2)와 같이 도출한 뒤에 이차방정식의 판별식 연산을 위한 식 (3)을 수행한다.

$$ax^2 + (b-s)x + (s \times x_{obj}) - y_{obj} + w = 0, Path_{obj} = Path_{sub},$$

$$s = \frac{Vy_{obj}}{Vx_{obj}}, w = 0.8, 0, -0.8 \quad (2)$$

w 는 차량의 폭을 보상하기 위한 값이며 중점(0,0)을 기준으로 차량의 폭만큼 예측경로에 +0.8, -0.8m씩 보상하여 더욱 강건한 판단을 할 수 있게 도와준다. 보상된 폭의 범위를 사용하여 $(x, y)_{obj}$ 가 가지는 $Y_{cnet, \min x, \min y}$ 가 음수인 경우 $Path_{sub}$ 의 w 는 -0.8, 양수인 경우 w 는 +0.8, 그 외의 경우는 w 는 0이 된다.

$$Disc = (b-s)^2 - 4a(s \times x_{obj} - y_{obj} + w),$$

$$\begin{aligned} &\text{if } Disc < 0, Cp(x, y) \text{ is none} \\ &\text{if } Disc = 0, one Cp(x, y) \\ &\text{if } Disc > 0, two Cp(x, y)'s \end{aligned} \quad (3)$$

$Disc$ 값이 0보다 작으면 $Cp(x, y)$ 가 발생하지 않기 때문에 충돌에 대해서 안전한 상황이며, $Disc$ 값이 0인 경우에는 $Cp(x, y)$ 가 하나만 발생, $Disc$ 값이 0보다 크면 두 개의 $Cp(x, y)$ 가 발생을 한다.

$Disc$ 값이 0보다 작은 경우를 제외한 나머지 두 경우는 $Cp(x, y)$ 가 발행하기 때문에 식 (4)를 이용하여 발생하는 $Cp(x, y)$ 를 사용하여 그 교점들이 실제로 부딪히는 $Cp(x, y)$ 인지를 판단해야 한다.

$$x_{cp1}, x_{cp2} = \frac{-(b-s) \pm \sqrt{Disc}}{2a},$$

$$\text{if } x_{cp} < -L, Cp(x, y) \text{ is none}, L = 4m \quad (4)$$

식 (4)에서 도출된 값 x_{cp1}, x_{cp2} 에 차량은 양수인 X 축 방향으로 이동하기 때문에 교점이 음수인 경우에는 위험한 순간이 아니지만 차량의 길이를 고려하여 실제로는 -4m보다 작은 교점의 위치부터 안전하다고 판단할 수 있게 된다. x_{cp1}, x_{cp2} 의 값이 둘 다 양수인 경우에는 두 값 중 원점에 가장 가까운 교점 $Cp(x, y)$ 를 사용한다. 이렇게 구한 하나의 교점 $Cp(x, y)$ 는 차량의 방향을 고려하지 않은 교점이기 때문에 $(x, y)_{head}$ 을 사용해서 실제로 부딪히는 점인지 아닌지 판단을 한다. x_{head} 은 $Path_{obj}$ 위에 존재하는 x_{obj} 의 방향 Vx_{obj} 에 따라 바뀌며, Vx_{obj} 가 양수이면 x_{obj}

를 기준으로 작은 임의의 값을 더해주며, Vx_{obj} 가 음수이면 x_{obj} 를 기준으로 작은 임의의 값을 빼준다. 본 위 연구에서는 임의의 작은 값 0.1을 사용 하였다.

$$\begin{aligned} &\text{if } Vx > 0, x_{head} = x_{obj} + 0.1 \\ &\text{if } Vx < 0, x_{head} = x_{obj} - 0.1 \end{aligned} \quad (5)$$

식 (5)로 구한 x_{head} 에 대한 y_{head} 는 식 (1)에 대입하여 구할 수 있게 되며, 두 점 $(x, y)_{head}$ 과 $(x, y)_{obj}$ 에 대해 각각 $Cp(x, y)$ 와의 거리를 통하여 식 (6)과 같이 실제로 사용 가능한 교점인지 판단할 수 있게 된다.

$$D_{head} = \sqrt{(x_{head} - x_{cp})^2 + (y_{head} - y_{cp})^2},$$

$$D_{obj} = \sqrt{(x_{obj} - x_{cp})^2 + (y_{obj} - y_{cp})^2} \quad (6)$$

$$\text{if } D_{head} < D_{obj}, Cp(x, y) \text{ is real point}$$

만약 D_{head} 의 값이 D_{obj} 의 값보다 작으면 $Cp(x, y)$ 은 실제로 부딪히게 되는 교점이 맞으며, D_{head} 의 값이 D_{obj} 의 값보다 크면 차량은 $Cp(x, y)$ 와는 반대로 향하게 되어 충돌을 하지 않게 되는 상황이 된다. 최종적으로 충돌의 위험이 있는 $(x, y)_{obj}$ 와 그 $Cp(x, y)$ 를 구하게 되었으면 Subject와 $Cp(x, y)$ 까지의 거리, Object와 $Cp(x, y)$ 까지의 거리와 함께, 각각의 속도를 구하여 충돌 예측 시간을 계산한다. Subject의 충돌 예측 시간을 구하기 위해서는 Subject와 $Cp(x, y)$ 까지의 거리 D_{sub} 를 식 (7)과 같이 구할 수 있게 된다.

$$D_{sub} = \int_0^{x_{cp}} \sqrt{4a^2x^2 + 4abx + b^2 + 1} dx \quad (7)$$

단 위와 같은 식은 20ms마다 연산을 해야 하는 시스템에 큰 부담을 주기 때문에 식 (8)을 사용한다.

$$D_{sub} = \sqrt{(x_{cp})^2 + (y_{cp})^2}, (x, y)_{sub} = (0, 0) \quad (8)$$

Subject의 D_{sub} 와 V_{sub} 를 통하여 Subject의 충돌 예측 시간 T_{sub} 와 함께 Object의 D_{obj} 와 V_{obj} 를 통하여 Object의 충돌 예측 시간 T_{obj} 을 식 (9)로 구할 수 있게 된다.

$$T_{sub} = \frac{D_{sub}}{V_{sub}} (m/s)$$

$$T_{obj} = \frac{D_{obj}}{(\sqrt{(Vx_{obj})^2 + (Vy_{obj})^2})} (m/s) \quad (9)$$

시험 차량을 기준으로 실차를 통하여 T_{obj} 와 T_{sub} 의 시간의 차로 차량의 위험을 판단한다. $|T_{sub} - T_{obj}| < 4(\text{sec})$ 를 만족하면서 $T_{sub} < 6(\text{sec})$ 를 만족하게 되는 시간이 발생하게 되면 Subject는 위험하다고 판단을 하여 차량을 정차하게 된다. 차량이 정차하게 된 이후 다시 차량을 출발시키기 위해서는 일정 구간내의 $Path_{sub}$ 상에 Subject와 충돌이 생기는 $Cp(x,y)$ 가 없다면 차량을 다시 출발시킨다.

3.2.2. 긴급 제동 기반 충돌 예측 알고리즘 개발

긴급 제동의 경우에는 Subject에 대한 현재 위치에서의 예측경로 $Path_{sub}$ 에서 차량의 속도 V_{sub} 에 비례하여 위험 구간을 설정하여, 범위 안에 $(x,y)_{obj}$ 가 갑자기 다가오는 경우에 바로 제동할 수 있게끔 알고리즘이 구성된다. $(x,y)_{obj}$ 에 대한 정보는 차량의 방향이 X축으로 양수방향이기 때문에 $(x,y)_{obj}$ 의 데이터 중에서도 가장 X축과 근접한 $(x,y)_{minx}$ 를 사용한다. Fig. 7에서 표현되는 $Path_{sub}$

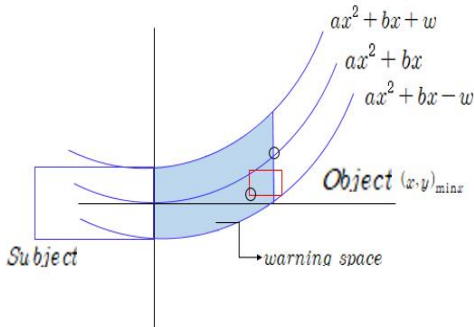
의 Y축 범위 내에 V_{sub} 의 속도에 비례한 X축 범위를 설정하여 위 범위 안에 $(x,y)_{minx}$ 의 존재 여부를 식 (10)으로 확인한다.

$$y_{range+} = Y_{minx} - (a * X_{minx}^2 + b * X_{minx}) + 1.6 < 0 \ \&$$

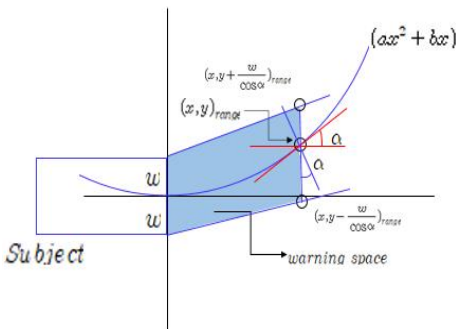
$$y_{range-} = Y_{minx} - (a * X_{minx}^2 + b * X_{minx}) - 1.6 > 0 \quad (10)$$

if $V_{sub} < 3(m/s), x_{range} = 8, \text{ if } V_{sub} > 3(m/s), x_{range} = 15$

V_{sub} 에 따라서 X축의 범위 x_{range} 는 최대 15m, 최소 8m, $Path_{sub}$ 에 따라 Y축의 범위 y_{range+}, y_{range-} 는 변동이 되며, 이 범위 안에 검출된 데이터 중 X축을 기준으로 가장 차량에 가까운 데이터 $(x,y)_{minx}$ 를 검출한다. 식 (10)으로 검출한 데이터 $(x,y)_{minx}$ 를 총 6회에 걸쳐 count하여 $(x,y)_{minx}$ 이 식 (10)의 범위에 포함된 누적 count를 고려하여 차량의 위험한 순간을 판단하여 차량을 제어하고, 위험한 순간을 판단하여 차량이 멈춘 뒤에는 다시 식 (10)을 통하여 범위 안에 $(x,y)_{minx}$ 의 검출이 없다면 다시 차량을 출발시킨다. 다만 위와 같은 식은 실제로 물체가 $Path_{sub}$ 에 존재하더라도 근집화된 점이 식 (10)의 범위 안에 포함되지 않는다면 차량 제어에 문제가 있어서 강건함을 이유로 다른 안으로 알고리즘을 설계하였다. 문제를 해결하기 위해서 근집화를 위하여 사용되는 물체에 검출된 근집화 되기 전의 $Rawdata((x,y)_{raw})$ 를 사용하며, Subject의 방향에 따른 범위의 보상을 위하여 V_{sub} 에 비례하여 증감하는 X축의 범위의 좌표 x_{range} 의 위치에 따른 기울기를 구하여 그 기울기에 따른 수직에 대한 각도 α 를 구한다. α 를 사용하여 Y축에 대한 범위를 $Path_{sub}$ 의 기울기에 따라 추가적으로 보상하여 차량의 폭 w 을 $w/\cos\alpha$ 로 보상하여 식 (11)과 같이 범위를 지정한다.



(a) Funnel for Path



(b) Collision Estimation

Fig. 7 Concept of Emergency Brake Algorithm

$$R_{+Y} = \left(\frac{y_{range+} + \frac{w}{\cos\alpha}}{x_{range}} \right) * x_{raw} - w < 0 \ \&$$

$$R_{-Y} = \left(\frac{y_{range-} - \frac{w}{\cos\alpha}}{x_{range}} \right) * x_{raw} + w > 0 \quad (11)$$

$$x_{range} = (V_{sub} * 2) + 10$$

위험 구역에 대한 범위는 이전과 달리 일차방정식으로 이뤄져있으며, x_{range} 의 범위 또한 최소 10m에서 최대 20m까지 확인하며, 해당 위험 구역 안에 $(x,y)_{raw}$ 가 검출이 되면 Subject 차량과 가까운 $(x,y)_{raw}$ 을 찾은 뒤에 그

점까지의 거리와 함께 Subject의 속도를 측정하여 충돌 예측 시간 T_{sub} 을 연산한다. 이렇게 구한 시간 T_{sub} 를 총 5회 count 하여 충돌 예측 시간이 4초 이내에 도달하게 되는 점 $(x,y)_{raw}$ 을 검출하면 차량은 위험하다고 판단하여 차량을 제어하며, 정지된 차량은 식 (11)을 통하여 검출된 $(x,y)_{raw}$ 가 없을 경우 다시 차량은 출발한다.

3.2.3. 협력 센싱 기반 충돌 예측 알고리즘 개발

앞에서 설명한 TTC와 긴급 제동 Algorithm의 경우 물체를 검출하기 위한 센서로 Lidar를 사용하였지만, Lidar의 한계로 인하여 센서가 검출할 수 없는 사각지대와 같은 곳의 Object는 검출을 할 수가 없다. 이를 보완하기 위하여 사각지대마다 인프라 카메라를 설치하여 사각지대에서 Lidar 센서로 검출 할 수 없는 Object에 대하여 인프라 카메라로 그 Object를 검출할 수 있게 하였다. 협력 센싱 기반 충돌 예측 Algorithm 또한 TTC Algorithm과 같은 기반이며, Lidar로 검출한 Object를 사용하는 TTC Algorithm과는 다르게 도로의 사각지대에 설치된 Infrastructure camera로 Object를 검출하여 이차식으로 구성된 Subject의 $Path_{sub}$ 에 일차식으로 구성된 $Path_{infra obj}$ 와의 교점 $Cp(x,y)$ 를 구해서 Subject와 Object가 그 $Cp(x,y)$ 까지 도착하기까지의 시간을 구하여 충돌의 예측을 찾아서 차량을 제어한다. Lidar로 검출할 수 없는 사각지대와 같은 경우는 사람 또한 보기 힘들기 때문에 사각지대마다 설치한 Infrastructure camera는 Lidar의 한계를 보완해 줄 수 있게 된다.

Infrastructure camera에서 검출한 Object는 절대 좌표인 위도와 경도 $(latitude, longitude)_{obj}$ 를 통신을 사용하여 Subject로 보낸다. Subject는 자신이 위치하고 있는 위도와 경도 $(latitude, longitude)_{sub}$ 를 통하여 상대 좌표에 맞춰서 Subject의 상대 좌표 (0,0)에 따른 Object의 상대 좌표 $(x,y)_{infra}$ 를 도출한다. Object의 예측 경로 $Path_{infra obj}$ 는 좌표 $(x,y)_{infra}$ 와 $(Vx, Vy)_{infra}$ 을 통해서 식 (12)로 도출된다.

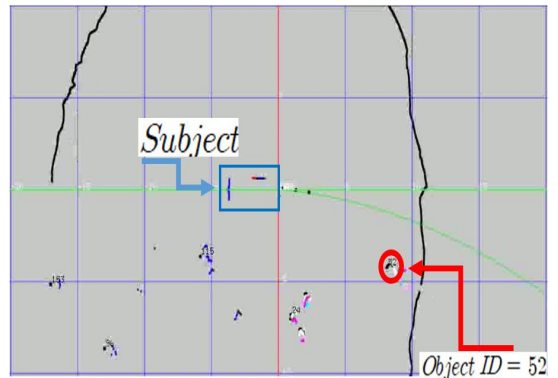
$$Path_{infra} = \frac{Vy_{infra}}{Vx_{infra}}(x - x_{infra}) + y_{infra} \quad (12)$$

$Path_{infra}$ 가 생성된 후에는 TTC Algorithm에서 설명한 과정을 식 (3)부터 식 (9)까지 그대로 반영한다.

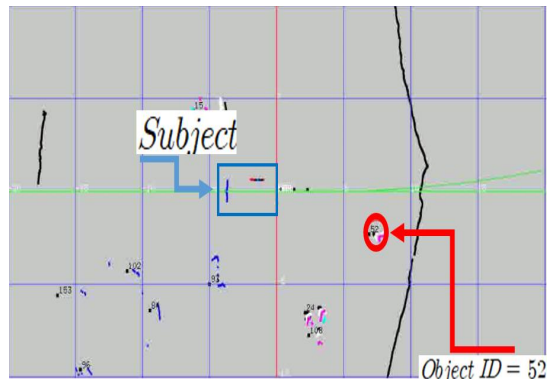
4. 시험 결과

4.1. TTC 기반 충돌 예측 알고리즘 시험 결과

실험은 실차를 통하여 이루어졌으며, 실제 차량을 운행하면서 임의의 위험한 순간을 만들기 위하여 차량이 움직이는 동안 사람이 차량에 부딪히는 상황에 대해 검증하였다. Fig. 8의 (a)는 중앙의 직사각형 표식은 Subject이며 우측 하단의 원형 표식은 Object를 표시한 것으로, TTC Algorithm이 판단한 Subject가 Object와 부딪힐 위험을 판단한 순간의 상황이며, Fig. 8의 (b)는 충돌이 예측되어 위험하다고 판단한 순간 이후로 완전 정지했을 때의 상황으로 완전히 정차한 순간이다. Subject가 정지한 이후 상대적으로 Object의 근처에 있는 상태가 되고, Fig. 9는 충돌 예측 시점부터 Subject가 완전 정차까지의 시간에 따른 Subject와 Object의 속도 변화를 나타낸 것이



(a) Before Warning State



(b) Warning State

Fig. 8 Test Scenario for TTC algorithm

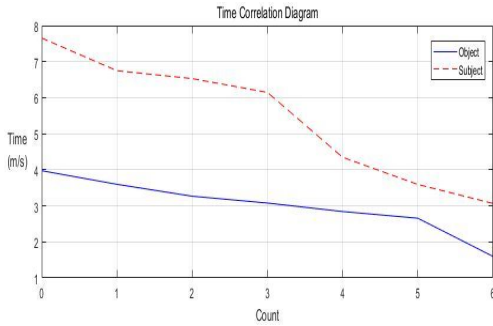


Fig. 9 Test Results for TTC Algorithm

다. 실선은 시간에 따른 Subject의 속도의 변화이고, 점선은 Subject가 감속하는 순간의 Object의 속도 변화이다. Table 2는 Fig. 8에서의 순간에 대한 데이터들의 값을 표시한 것이다.

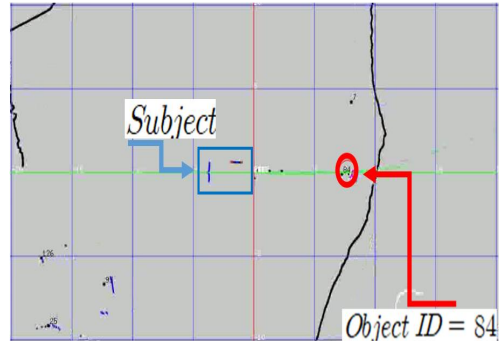
Table 2 Data for TTC Algorithm Test Results

Variable	Value
$Path_{sub} = ax^2 + bx$	$-0.0571x^2 - 0.0571x$
V_{sub}	1.938889(m/s)
Object ID	52, warning = $(x, y)_{cent}$
$(x, y)_{cent}$	(8.82, -4.3)
$(x, y)_{minx}$	(8.6, -4.21)
$(x, y)_{miny}$	(8.86, -4.46)
$(Vx, Vy)_{cent}$	(-0.225, 0.8)
$Cp(x, y)$	(8.180537, -4.2883)
D_{obj}	2.361852
D_{sub}	8.427772
T_{obj}	2.842051
T_{sub}	4.346702

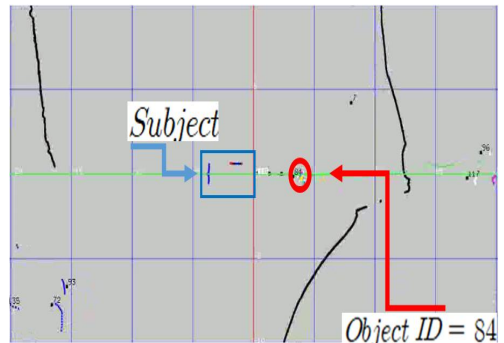
Table 2를 확인하게 되면 Fig. 8에서 충돌을 예측한 순간의 Subject의 위치는 (0,0), Object의 위치는 (8.82, -4.3), Subject의 예측 경로 상에서 두 물체의 충돌 예측 지점은 (8.180537, -4.2883)이며, Subject의 속도는 1.9389(m/s), Object의 속도는 0.8003(m/s), Subject가 충돌 예측 지점까지 도달하는 시간은 4.346702(sec), Object가 충돌 예측 지점까지 도달하는 시간은 2.84251(sec)로 TTC Algorithm에서 $|T_{sub} - T_{obj}| < 4(sec)$ 와 $T_{sub} < 6(sec)$ 를 만족하여 Subject인 차량은 정지를 실행하였음을 확인할 수 있게 된다.

4.2. 긴급 제동 기반 충돌 예측 알고리즘 시험 결과

긴급 제동 충돌 예측 알고리즘의 실험 결과는 Fig. 10에 나타내었다. Fig. 10의 (a)는 기존의 긴급 제동에 대하여 Subject가 긴급 제동 Algorithm을 통해 Object가 다가오는 순간의 시나리오이다. Fig. 10의 (b)는 Subject가 완전 정지한 순간의 시나리오 화면이다. Table 3은



(a) Before Warning State



(b) Warning State

Fig. 10 Test Scenario for Emergency Brake algorithm

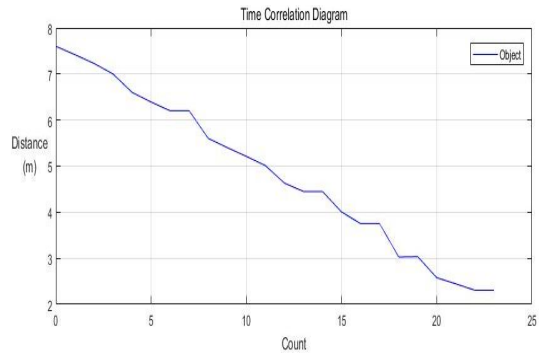


Fig. 11 Test Results for Emergency Brake algorithm

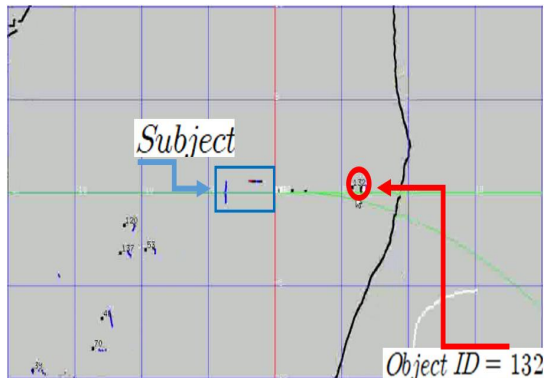
Table 3 Data for Emergency Brake algorithm

Variable	Value
V_{sub}	1.913889(m/s)
Object ID	84
$(x,y)_{minx}$	(7.6, -0.19)

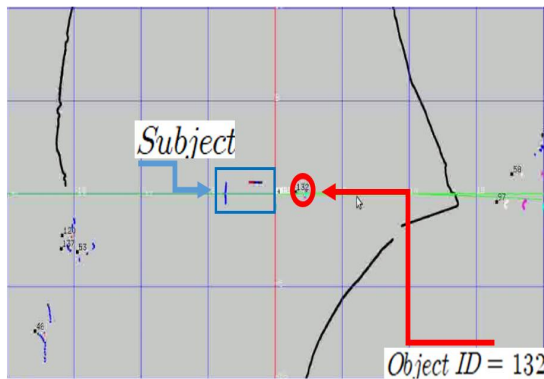
Fig. 11에서 충돌을 예측한 순간의 데이터들을 표시한 결과이다.

Table 3에서 Subject의 위치는 (0,0)일 때 Object의 위치는(7.6, -0.19)이며 차량의 긴급 제동 Algorithm에 따라 Object가 Subject의 전방 기준 15m 이내에 들어온 순간부터 count를 시작하여 정지를 실시한 모습을 확인할 수 있게 된다.

위에서 적용한 Algorithm에 단점인 위험 감지 범위 외에 근집화된 Object가 발생한 경우에 대한 개선된 긴급 제동 Algorithm의 실험 결과는 아래에 Fig. 12에 개선된



(a) Before Warning State



(b) Warning State

Fig. 12 Test for Improved Emergency Brake algorithm

긴급 제동을 통하여 Subject가 충돌을 예측한 순간에 대한 그림이다.

Fig. 13은 개선된 긴급 제동 알고리즘의 결과를 보여주고 있다. Table 4는 Subject가 충돌을 예측한 순간에 대한 데이터들의 집합이다.

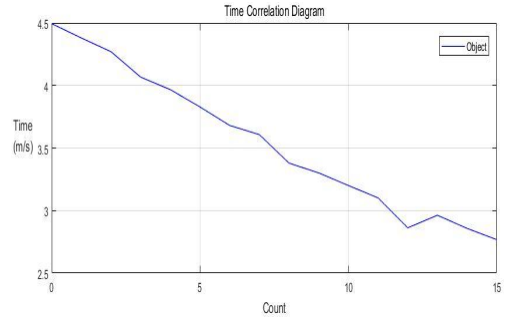


Fig. 13 Test Results for Improved Emergency Brake algorithm

Table 4 Data for Improved Emergency Brake algorithm

Variable	Value
$Path_{sub} = ax^2 + bx$	$-0.01791x^2 + 0.045832x$
V_{sub}	1.675(m/s)
$Cp(x,y)$	(6.62, 0.34)
D_{sub}	6.628725
T_{sub}	3.957447

Table 4에서 Subject의 위치는 (0,0)일 때 Object의 위치는(6.62, 0.34)이며 개선된 긴급 제동 Algorithm에 따라 Subject가 Object까지의 거리 6.628725(m)까지의 충돌 예측 시간이 3.957447(sec)이기 때문에 설정한 충돌 예측 시간 범위인 4초 이내의 범위에 들어서 Subject의 완전 정지를 실행함을 확인할 수 있게 된다.

4.3. 협력 센싱 기반 충돌 예측 알고리즘 시험 결과

Fig. 14는 경로 상에 Lidar로 검출되지 않은 Object에 대한 정보를 인프라 카메라를 통하여 자기 차량이 TTC Algorithm을 통하여 Object와의 충돌 예측 상황에 대한 순간에 대한 그림이다. 실선은 자기 차량의 예측 경로이고 점선은 Object의 예측 경로이며, 두 경로 상에 존재하는 원형 표시는 Subject와 Object 차량의 충돌 예측 위치이다. Fig. 15는 충돌을 예측한 순간으로부터 자차 차량이 완전 정지하기까지의 속도의 변화량을 나타낸 것이다.

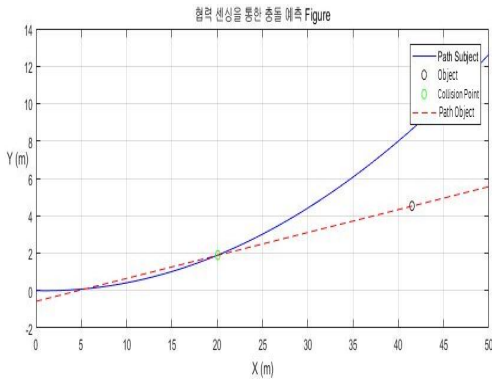


Fig. 14 Test Vehicle

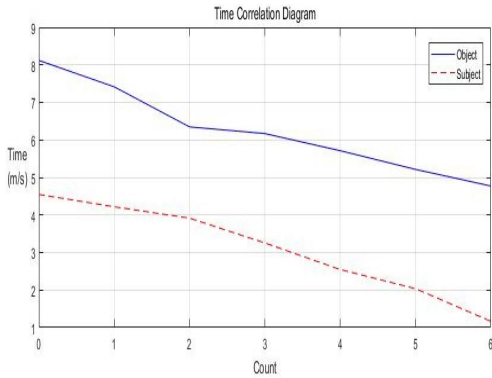


Fig. 15 Test Vehicle

실선은 Subject의 속도 변화, 점선은 Object의 속도 변화를 나타낸다. Table 5는 Subject가 충돌을 예측한 시점의 데이터를 보여주고 있다.

Table 5에서 Subject의 위치 (0,0)을 기준으로 Object

Table 5 Data for prediction time

Variable	Value
$Path_{sub} = ax^2 + bx$	$0.00528x^2 - 0.011438x$
V_{sub}	3.53552(m/s)
$(x, y)_{infra}$	(41.516, 4.5285)
$(Vx, Vy)_{infra}$	(-8.423, -1.0334)
$Cp(x, y)$	(20.0711, 1.8975)
D_{obj}	21.6057
D_{sub}	20.1976
T_{obj}	2.545991
T_{sub}	5.712748

의 위치는 (41.516, 4.5285)이며, 충돌 예측 지점 (20.0711, 1.8975)를 기준으로 Subject가 충돌 예측 지점까지의 거리는 20.1976(m)이며, Object가 충돌 예측 지점까지의 거리는 21.6057(m), Subject가 충돌 예측 지점까지 도달하는 시간은 5.712748(sec)이며 Object가 충돌 예측 지점까지 도달하는 시간은 2.545991(sec)로 TTC Algorithm에서 $|T_{sub} - T_{obj}| < 4(sec)$ 와 $T_{sub} < 6(sec)$ 를 만족하여 Subject인 차량은 정지를 실행하였음을 확인할 수 있게 된다.

6. 결론

국내 최초로 인프라에 설치된 카메라로부터 전달받은 차량 및 보행자 객체 정보와 기존 환경 센서 정보와의 융합을 통해 교차로 및 유턴 등 복잡한 상황 하에서도 자율주행이 수행될 수 있는 융합 충돌 회피 알고리즘을 개발하였다. 개발된 알고리즘을 검증하기 위해 자율주행서플버스에 알고리즘을 탑재하고 성능을 검증하였다. 기존에 많은 연구를 수행했던 TTC기반 알고리즘의 새로운 제안, 긴급 제동 알고리즘의 개선 등을 수행하였으며 중점적으로 국내 최초인프라 정보를 입력 받아 환경 센서 정보와의 융합을 통한 협력 센싱 Algorithm을 이용한 위험 판단 알고리즘을 추가함으로써 차량과 보행자가 혼재된 혼합류 상황에서의 대처가 가능하며 유턴, 교차로 등 복잡한 환경에서의 자율주행이 가능함을 실증하였다. 그러나 본 연구는 20kph 미만의 저속 상황에 대한 검증으로 자율주행 서플버스의 대용화를 위해 고속에서의 협력 센싱 기반 알고리즘 개발에 대한 알고리즘에 대한 연구가 필요할 것으로 판단된다.

본 연구의 결과를 요약하면 다음과 같다.

- 1) 순환식 서플버스의 충돌 예측 알고리즘의 아키텍처를 수립하였다.
- 2) 충돌 예측은 기존 연구에서 개발 적용되었던 TTC (Time to Collision) 알고리즘과 긴급제동 알고리즘 뿐만 아니라 협력 센싱 정보를 이용한 충돌 알고리즘을 추가하여 차량과 보행자 등이 혼재된 혼합류 상황 하에서도 자율주행 4단계가 가능한 자율주행 자동차의 충돌 예측 알고리즘을 개발하였다.
- 3) 개발된 알고리즘에 대해 운행 중인 실제 교내 서플버스를 이용하여 주행 경로 상의 혼합류 상황에서 유효성을 검증하였다.

4) 그러나 본 연구는 20kph 미만의 저속 상황에 대한 검증으로 자율주행 셔틀버스의 대중화를 위해 고속에서의 협력 센싱 기반 알고리즘 개발 및 다양한 환경에서의 추가 대한 알고리즘에 대한 연구가 필요할 것으로 판단된다.

후 기

이 논문은 2019년도 정부(경찰청)의 재원으로 도로교통공단의 지원을 받아 수행된 연구임(No.POLICE-L-00001-01-101, 자율주행차의 도로주행을 위한 운행체계 및 교통인프라 연구개발)이며 2019년도 정부(경찰청)의 재원으로 도로교통공단의 지원을 받아 수행된 연구임(POLICE-L-00003-01-101, IoT 기반 교통안전시설 정보제공 및 운영관리 기술개발)임. 또한, 이 논문은 국토교통부 도심도로 자율협력주행 안전·인프라 연구 사업의 연구비지원(과제번호 19PQOW-B152473-01)에 의해 수행되었음.

참고문헌

- (1) 민경찬, 이명수, 2015, “자율주행자동차의 국내외 임시운행허가 및 안전기준 개발 동향”, 오토저널, Vol. 3, No. 9, pp. 53~58.
- (2) 최인성, 민경찬, 홍윤석, 김규현, 2015, “자율주행 자동차 도로운행 관련 국내외 법규 현황”, 한국자동차공학회 부문종합 학술대회, Vol. 2015, No. 5, pp. 485~485.
- (3) 민경찬, 이진복, 이경수, 2004, “차량 주행안정성 제어시스템의 자동안전 로직”, 대한기계학회논문집, Vol. 28, No. 11, pp. 1685~1691.
- (4) 김범준, 정용환, 채홍석, 민경찬, 이경수, 2016, “센서융합 통합환경인지를 통한 도심자율주행 알고리즘 개발”, 대한기계학회 춘추학술대회, Vol. 2016, No. 12, pp. 1313~1318.
- (5) Donghwi Lee, Kwangjin Han, Sangmin Cho, Yongsun Kim and Kunsu Huh, 2012, “Development of a Frontal Collision Detection Algorithm Using Laser Scanners”, Transaction of the Korean Society of Automotive Engineers, Vol. 20, No. 3, pp. 113~118.
- (6) Dongjin Yoon, Jaehwan Kim and Jungha Kim, 2012, “LiDAR Point Cloud Data Clustering and Classification for Obstacle Recognition of UGV”, 한국자동차공학회 추계학술대회 및 전시회, pp. 1339~1343.
- (7) Keonyup Chu, Jaehyun Han, Minchae Lee, Dongchul Kim, Kichun Jo, Dong-eon Oh, Enae Yoon, Myeong-gi Gwak, Kwangjin Han, Donghwi Lee, Byungdo Choe, Yangsoo Kim, Kangyoon Lee, Kunsu Huh and Myoung-ho Sunwoo, 2011, “Development of an Autonomous Vehicle: AI”, Transaction of the Korean Society of Automotive Engineers, Vol. 19, No. 4, pp. 146~154.
- (8) Donghoon Shin, Kangmun Park, Manbok Park, 2018, “Effects of Vehicular Communication on Risk Assessment in Automated Driving Vehicle”, applied science, volume 8 issue 12.
- (9) 이호준, 채홍석, 서호태, 이경수, 2018, “자율주행을 위한 레이더 기반 인지 알고리즘의 정량적 분석”, 자동차안전학회지, Vol. 10, No. 2, pp. 29~35.
- (10) 서호태, 박성렬, 이경수, 2017, “차선 유실구간 측위를 위한 레이저 스캐너 기반 고정 장애물 탐지 알고리즘 개발”, 자동차안전학회지, Vol. 9, No. 3, pp. 24~30.