

# 기상 환경을 고려한 자율주행 차량용 교통 시뮬레이션에 관한 연구

## A Study on the Traffic Simulation for Autonomous Vehicles Considering Weather Environment

이서영 · 용성중 · 박효경 · 유연휘 · 문일영\*  
한국기술교육대학교 컴퓨터공학과

Seo-Young Lee · Sung-Jung Yong · Hyo-Gyeong Park · Yeon-Hwi You · Il-Young Moon\*  
Department of Computer Engineering Korea University of Technology and Education, Cheonan 31253, Korea

### [요 약]

자율주행 차량의 개발은 현재 각종 기업체와 연구소에서 활발하게 이뤄지고 있다. 특정 산업뿐만 아니라 일상생활 속 상용화에 대한 기대감 역시 높아진 상태이다. 자율주행 차량을 위한 시뮬레이터는 안정성 및 비용을 고려했을 때, 알고리즘 개발 및 수행에 있어서 필수 요소이다. 이러한 필요 속에서 다양한 시뮬레이터 및 시뮬레이터용 플랫폼들이 등장하고 있지만, 현실 세계의 다양한 기상 환경 요소를 반영한 시뮬레이션에 대한 연구는 아직 미흡한 편이다. 본 논문은 기상 환경을 고려할 수 있는 자율주행 차량용 교통 시뮬레이션을 제안하였다. 설정할 수 있는 기상 환경을 크게 4가지로 분류하였고, 이를 적용할 수 있는 개선된 충돌 방지 알고리즘을 제시한다. 시뮬레이션 개발은 자율주행을 위한 개발 도구 CARLA의 Python API를 통해 이루어졌고, 기존 충돌 알고리즘과의 수행 결과를 비교하였다. 이를 통해 실생활의 다양한 기상 환경 요소를 반영할 수 있는 고도화된 자율주행 차량용 시뮬레이션 개발을 위한 개선점을 제안하고자 하였다.

### [Abstract]

The development of autonomous vehicles are currently being actively carried out by various companies and research institutes. Expectations for commercialization in daily life as well as specific industries are also rising. Simulators for autonomous vehicles are an essential element in algorithm development and execution considering stability and cost. In this need, various simulators and platforms for simulators are emerging, but research on simulations that reflect various meteorological environmental factors in the real world is still insufficient. This paper proposes a traffic simulation for autonomous vehicles that can consider the weather environment. The weather environment that can be set is largely classified into four categories, and an improved collision prevention algorithm to apply them is presented. Simulation development was conducted through Carla's Python API, a development tool for autonomous driving, and the performance results were compared with existing collision algorithms. Through this, we tried to propose improvements for the development of advanced self-driving vehicle simulations that can reflect various weather environmental factors in real life.

**Key word** : Autonomous Driving, Collision Avoidance, Motion Planning, Traffic Simulation.

<http://dx.doi.org/10.12673/jant.2023.27.1.36>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 27 December 2022; Revised 31 January 2023  
Accepted (Publication) 21 February 2023 (28 February 2023)

\*Corresponding Author; Il-Young Moon

Tel: +82-041-560-1493

E-mail: iymoon@koreatech.ac.kr

# 1. 서론

자율주행은 미래의 꿈이 아니라 현실이 되고 있으며, 연구소와 기업들의 자율주행 모빌리티 개발 및 상용화를 위한 연구가 수행되고 있다. 이러한 자율주행기술은 교통 혼잡 감소, 오염 감소, 운송 비용 절감 등 새로운 도로 및 기반 시설 비용 감소의 결과를 가져올 것이다[1].

자율주행 모빌리티는 21세기 우리 사회, 경제, 환경에 큰 영향을 미칠 수 있는 새로운 기술이며, 자율주행 모빌리티 개발과 관련된 기술적 및 과학적 과제는 현재 많은 자동차 회사와 연구소에서의 주요 관심 분야이다. 그러나 완전한 자율주행 모빌리티를 구현하기 위해서는 많은 법적, 기술적, 알고리즘적인 문제들이 존재한다. 현재 출시된 자율주행 기능은 아직 운전 상황을 감시해야 하는 제약조건이 있는 자율주행인 상태이다[2]. 인간이 제어하는 차량은 향후 수십 년 동안은 여전히 도로에 남아 있을 가능성이 크며, 자율주행 모빌리티와 함께 미래의 교통 환경을 공유할 것이다[3]. 이러한 혼합 환경 속에서 자율주행 모빌리티의 운전 정책 결정과 협상, 보행자와 차량 등의 장애물과의 충돌 회피와 관련한 기술들은 필수적일 것이다. 현실의 주행 환경은 매우 복잡하고 통제할 수 없으며, 실시간으로 속도 및 방향의 개선이 필요하다. 때문에 빠르고 효율적으로 다루기 위한 여러 구조와 방법론은 자율주행 알고리즘의 지속적인 연구 과제이다.

자율주행 연구에서 기상 환경을 고려하거나 기상 요소의 영향력을 다루는 비중은 상대적으로 적은 편이다. 관련 연구 동향으로는 인지(Perception) 영역에서 기상 요소가 라이다(Lidar)나 레이더(Rader)와 같은 자율주행 센서에 미치는 영향을 다루는 연구가 있다[4]. 또 운전자가 수동으로 차량을 제어하는 조건부 자율주행과 관련하여 운전자의 나이와 날씨, 환경 등이 제어권 전환에 미치는 영향과 관련한 연구 등이 진행 중이다[5]. 그러나 자율 주행 알고리즘의 성능에 영향을 주는 핵심 단계인 모션 플래닝의 전통적인 고려요소는 지형물, 도로 규칙, 인접 차량, 장애물 등으로 날씨와 같은 기상 환경은 잘 반영되지 않는 것 같다. 기상 환경은 일차적으로 도로 환경에 큰 영향을 주며 운전 환경에도 영향을 미치기에 자율주행 상황에서 충돌 회피를 결정하는 알고리즘 개발에 있어서 이를 고려한 연구 및 시뮬레이션이 필요할 것이다[6],[7].

본 연구에서는 연구에 필요한 이론적 배경과 기존 연구를 통해 자율주행 모빌리티의 충돌 회피 알고리즘을 살펴보고, 장애물의 종류 및 기상 환경에 따라 모빌리티의 행동을 계획하는 충돌 회피 알고리즘을 개선하였다. 또한, 자율주행용 오픈소스 시뮬레이션 개발 도구인 CARLA (Car Learning to Act)를 활용하여 도시 교통 환경 및 기상 환경을 적용할 수 있는 시뮬레이션을 수행하였다[8]. 기존 알고리즘 및 개선된 알고리즘을 해당 시뮬레이션 환경에서 실험하여 결과를 비교하여 분석하며, 이를 토대로 기상 환경을 고려한 고도화된 충돌 회피 알고리즘 및 시뮬레이션 개발을 목표로하고자 한다.

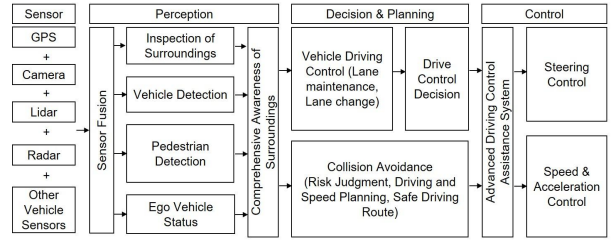


그림 1. 자율 주행 알고리즘의 구조  
Fig. 1. Architecture of autonomous driving algorithm

## II. 이론적 배경

### 2-1 자율주행 알고리즘의 구조

보편적인 자율주행 시스템의 구조는 크게 그림 1과 같은 4 단계로 이루어진다[9]. 자율주행 차량에 부착된 GPS, Camera, Lidar, Radar 등의 다양한 센서로부터 주변 환경의 데이터를 수집한 후 이를 융합하여(센서 퓨전) 데이터를 해석하는 인지 단계를 거친다. 인지 과정에서는 주변 차량, 보행자, 교통 신호, 현재 위치, 진행방향지시등의 종합적인 상황이 포함된다. 세 번째는 판단 및 계획 과정으로 앞 단계에서 인지된 종합적 인지로부터 실질적으로 차량의 주행을 제어한다. 차선변경, 경로 결정, 속도 결정 등이 이때 이루어지고 위험을 판단하여 주행 경로를 계획하는 것 역시 이 단계에서 이루어진다. 마지막 4단계는 제어 단계로서 앞서서 판단하고 계획한 정보를 토대로 실제로 차량의 회전, 정지, 감속, 가속 등을 제어한다.

### 2-2 Time to Collision (TTC, 충돌시간)

TTC는 차체와 장애물 간의 충돌까지 남은 시간을 의미하며 실제 주행 상황에서 센서 측정값을 통해 다음 수식 (1)과 같이 계산한다.

$$TTC = \frac{(x_1 - x_f)}{|v_1 - v_f|} \quad (1)$$

이 때,  $x_1$ 은 선두차량 위치,  $x_f$ 는 후위차량 위치,  $v_1$ 은 선두차량 주행속도,  $v_f$ 는 후위차량 주행속도를 의미한다.

감속 혹은 제동 시점을 위한 결정에 TTC를 사용하기 때문에 정확한 TTC 산출은 매우 중요하다. TTC를 구하기 위한 대표적인 접근 방법으로는 시뮬레이션 기반 접근법과 추정 기반 접근 방식이 있다. 시뮬레이션 기반 접근의 주요 특징은 각 시간 단계에서 차량의 위치가 계산되며, 충돌 확인이 수행된다. 추정적 접근 방식은 바운딩 박스(bounding box)와 같은 간단한 기하적 구조 기반으로 위치를 추정하고 고정된 속도를 가정한다. 정확도 측면에서는 시뮬레이션 기반 접근의 TTC 계산이 우수하지만, 실시간 어플리케이션에서 계산 속도를 고려하여 대부분 추

정적 접근으로 TTC를 계산한다.

### 2-3 충돌 회피 알고리즘

효율적인 경로 생성을 위해 모션 플래닝은 핵심적인 단계이다. 모션 플래닝 업무의 복잡성을 반영하여 만들어진 계층적 구조(hierarchical architecture)는 자주 사용되는 구조이다. 계층적 구조는 일반적으로 광역 경로 계획부(route planning), 행동 결정부(behavioural planning), 지역적 경로 계획부(local planning)의 계층으로 구성되며 계층구조의 상위로 갈수록 추상화의 정도가 높아진다.

광역 경로 계획부에서는 도로망을 통해 초기 위치에서 목적지까지 넓은 범위에서의 경로 생성을 다룬다. 주로 사용되는 방법으로는 A\*(A star) 알고리즘과 다익스트라(Dijkstra) 최단 경로 알고리즘이 있다. 행동 결정부는 거리 유지, 차선 변경, 인접 차량과의 상호 작용 등과 같은 주행에 대한 결정을 수행하며 크게 finite state machine (FSM) 이라고도 불리는 규칙 기반 제어(rule-based controlling) 방법과 마르코프 결정 과정(markov decision process) 방법 등의 접근이 있다[10],[11]. 기존에는 정교한 시나리오를 바탕으로 하는 규칙 기반 제어 방법이 주로 사용되었지만, 최근에는 인공 지능 기술의 발달과 함께 강화학습(reinforcement learning) 기반의 접근들도 활발하게 연구되고 있다. 이는 방대한 데이터를 통해 주어진 환경에서 에이전트가 액션에 따른 보상(reward)을 최대화 하는 방법을 찾도록 학습한다. 지역적 경로 계획부에서는 앞선 단계에서 생성된 경로가 센서에 의해 감지된 장애물과의 충돌을 방지하면서 상황에 따라 가변적인 경로를 생성하는 역할을 수행한다[12],[13]. 본 논문에서 제안하는 충돌 회피 알고리즘의 범위는 지역적 경로 계획부 수준이며 규칙 기반 제어 방식을 사용한다고 볼 수 있다.

## III. 연구방법

### 3-1 모션 플래닝을 위한 시스템 아키텍처

그림 2는 행동 계획을 위한 알고리즘의 전반적인 아키텍처를 보여주고 있다. 먼저 인근에 있는 물체에 대한 정보를 받아온다. 이때 물체는 보행자와 차량, 장애물로 분류된다. 다음으로 차량과 물체 사이의 거리를 불러오고, 물체가 속도를 가지고 있다면 속도를 확인하여 이를 바탕으로 장애물까지의 충돌 시간(Time To Collision)을 계산한다. 이 때 TTC는 추정 기반 접근 방식으로 차량 위치를 차량 경계 상자의 중심으로 설정하여 계산하는 방식을 사용했다. 그 후 기상 환경을 인식하여 인식된 정보들을 종합하여 차량의 속도를 조절하여 차량의 충돌을 방지할 수 있도록 향후 행동을 계획한다. 계획된 해당 정보는 로컬 플래너(local planner) 모듈로 전달되며 향후 로컬 플래너에서 조향각(steering), 스로틀(throttle), 브레이크(brake)를 제동하여 방향 및 속도를 제어한다.

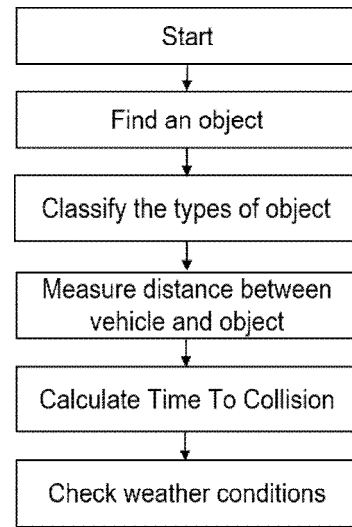


그림 2. 모션 플래닝을 위한 시스템 구조  
Fig. 2. System architecture for motion planning

### 3-2 정지 알고리즘

그림 3은 충돌 회피를 위한 정지 알고리즘의 흐름도를 보여준다. 먼저 센서를 통해 감지된 일정 거리 안에 있는 물체가 감지되어 저장된다. 객체의 종류는 보행자, 인접 차량, 장애물로 분류되고 각 타입에 따라 안전거리(safety distance)를 결정한다. 해당 안전거리 이내로 인식된 객체가 들어온다면 차량은 브레이크를 작동하여 충돌을 방지하도록 한다.

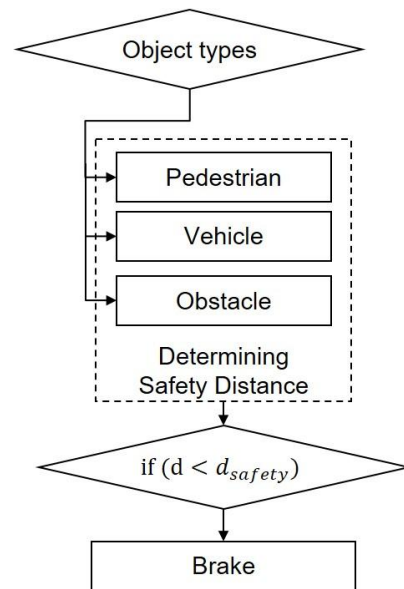


그림 3. 충돌 회피를 위한 정지 알고리즘  
Fig. 3. Braking algorithm for collision avoidance



그림 4. 시뮬레이션 환경에서의 기상 환경, 좌측부터 차례로 맑은 정오, 비 오는 정오, 습윤하고 구름 낀 정오, 맑은 저녁

Fig. 4. Weather conditions in simulation environment clear noon, mid rainy noon, wet cloudy noon, clear night from the left to the right

표 1. 날씨 환경에 대한 가시거리와 최소 근접 경계 가중치  
Table 1. Visibility and min proximity threshold weight for weather conditions

Weather condition	Visibility	Min proximity threshold weight
Clear Night	150 m	6.67
Clear Noon	1000 m	1.00
Mid Rainy Noon	400 m	2.50
Wet Cloudy Noon	100 m	10.00

### 3-3 기상 환경을 고려한 속도 계획 알고리즘

본 연구가 제안하는 충돌 회피 알고리즘은 각 기상 환경에 상응하는 가시거리에 따라 표 1과 같은 상대적인 가중치를 부여하여 해당 값에 영향을 주도록 하였다. 이는 차량의 최소 근접 임계 값(min proximity threshold)에 변화를 준다.

최소 근접 임계 값은 차량의 충돌 방지를 위해 다른 차량과 유지해야 하는 경계 값을 의미하며 이는 차량의 속도 계획에 영향을 미치는 요소로 작용한다.

가시거리가 가장 긴 맑은 정오(Clear Noon) 환경의 최소 근접 경계 값의 가중치를 기본 값인 1로 설정하였고, 가시거리가 짧은 다른 기상 환경들은 라이다, 레이더 등의 센서를 통한 주변 환경 인식에 악영향을 미치며 사고 위험률이 높을 것이라는 가정을 가지고 가시거리에 반비례하는 가중치 값을 설정하였다.

### 3-4 제어기

방향과 감속, 가속, 정지 등의 제어(control)를 위해서는 모 바일 로봇, 드론 등의 자율 주행체에서 널리 사용되는 PID (proportional-integral-derivative) 제어를 사용하였다. 제어량 계산식은 수식 (2)와 같다.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \dot{e}(t) \quad (2)$$

여기서  $u(t)$ 는 시간에 따른 제어량,  $e(t)$ 는 측정값과 제어의 목표값 사이의 차이값(편차)을 의미하며  $K_p$ 는 비례 제어,  $K_i$ 는 적분 제어,  $K_d$ 는 편차 제어를 조절하는 계수이다. PID 제어는 출력값이 다시 입력에 영향을 미치는 피드백(feedback) 방식의 제어 기법이다.

## IV. 시뮬레이션 및 연구결과

### 4-1 실험 설계

기상 조건으로는 도심 교통 환경 속에서 일상적으로 경험할 수 있으면서 낮과 밤, 비, 구름 등의 여러 환경 변화를 주고자 하였다. 이에 Carla Python Client API 0.9.13 버전에서 사전 결정된(preset) 14개의 기상 목록 중 맑은 밤, 맑은 정오, 비 오는 정오, 습윤하고 구름 낀 정오의 4가지 환경 조건을 선정하였고, 기존 연구를 참고하여 표 1과 같이 가시거리 값을 지정하였다 [14],[15].

실험의 종속 변수는 표 2와 같이 주행 시간과 충돌 횟수로 설정하였다. 주행 시간은 주어진 지도 상의 경로를 한 바퀴 도는데 걸리는 시간을 초(s) 단위로 측정하기로 하였다. 충돌은 차량, 보행자, 인근 지형물과의 총 충돌 횟수를 의미하며 1회(count)를 단위로 하였다.

실험은 기존의 Carla 알고리즘과 본 연구가 제안하는 가시거리에 따라 근접 임계 값의 가중치를 변경하여 주행하도록 하는 알고리즘을 각 기상 조건에 따라, 교통(traffic)의 여부를 변화로 주어 수행하였다. 각 경우에 대해 주행 시간, 충돌 횟수를 측정하였으며, 교통량은 시뮬레이션에서 제공하는 기능을 사용하여 30개의 차량과 8명의 보행자가 주어진 가상 환경 안의 임의 위치에 배치되어 이동하도록 하였다

### 4-2 실험 환경

시뮬레이션 구축 및 실험은 자율주행 시스템의 개발을 위해 만들어진 CARLA 오픈소스 라이브러리에서 제공하는 Python API를 사용하였고, 표 3과 같은 환경에서 수행하였다.

표 2. 실험에서 설정한 종속 변수  
Table 2. Overview of the dependent variables

Dependent variables	Unit
driving time	s
collisions and critical encounter	count

표 3. 실험 환경  
Table 3. Experiment environment

OS	CPU	RAM	GPU
Window 10	Intel(R) Core(TM) i7-7700	32.0 GB	NVIDIA GTX 1060 6GB

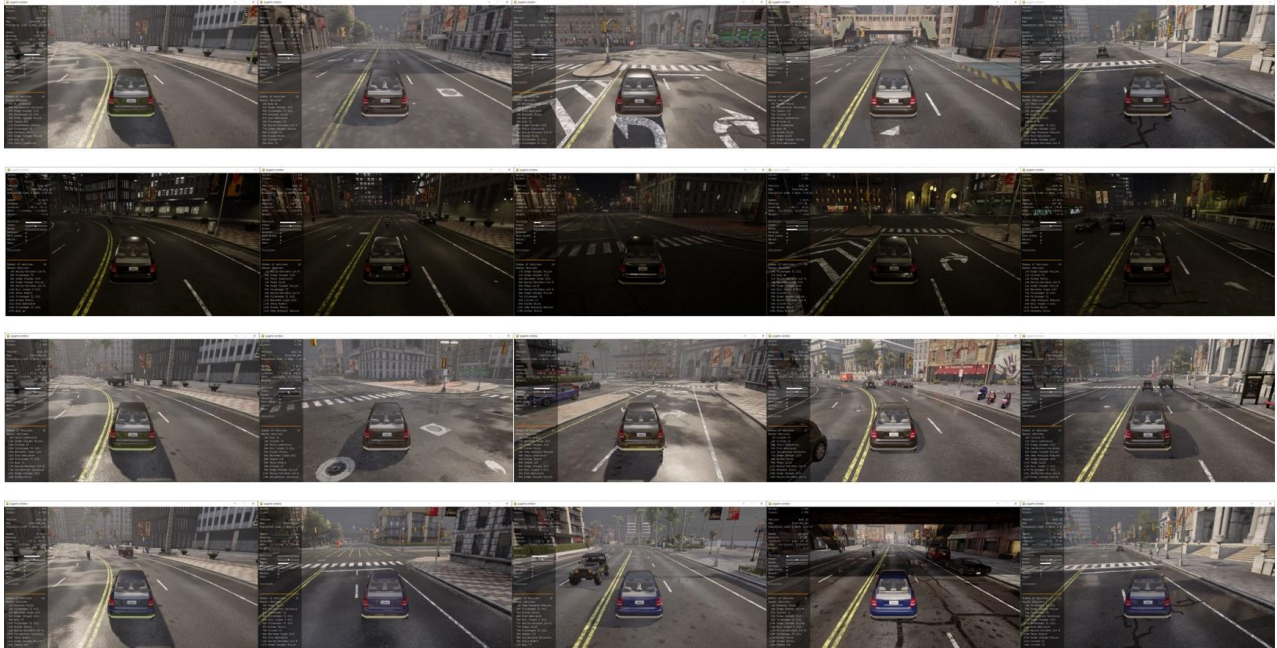


그림 5. 각 기상 환경에 대한 개선된 알고리즘에서의 주행 시나리오 장면, 위에서부터 차례로 맑은 정오, 맑은 저녁, 습윤하고 구름 낀 정오, 비 오는 정오

Fig. 5. Scenes of driving scenarios in improved algorithms for each weather environment, from the top, clear noon, clear evening, wet and cloudy noon, rainy noon.

### 4-3 실험 결과

표 4는 각 기상 환경 및 교통량에 따라 기존 알고리즘과 개선된 알고리즘의 결과를 보여준다. 표에서 제시한 기존 알고리즘은 Carla 환경에서 Github를 통해 오픈소스로 공개한 기본 주행(navigation) 알고리즘이다[16]. 개선된 알고리즘은 논문에서 이전 장을 통해 제안한 알고리즘을 의미한다. 또한 C는 맑은 정오(clear noon), CN은 맑은 저녁(clear night), MR은 비 오는 정오(mid noon rainy), WC는 습윤하고 구름 낀 정오(wet cloudy noon) 환경을, T는 교통량(traffic)을 적용한 환경을 나타낸다.

전반적으로 교통량을 적용했을 때, 신호 대기 전 후방 객체(보행자, 상대 차량)에 따른 속도 조절(감속)의 영향으로 인해 주행 시간이 길어지는 경향성을 보인다.

날씨 변화에 따른 주행 시간은 일정한 규칙을 찾기 어려운 것으로 여겨진다. 다만, 연구에서 제안한 알고리즘의 경우 날씨 변화에 따라 최소 근접 경계 거리의 증가가 주행 시간의 증가에 영향을 주었을 것이라고 예상할 수 있다. 충돌 횟수의 경우 표 5와 같이 개선된 알고리즘에서 줄어든 결과를 얻을 수 있었다. 그러나 실험 횟수가 적다는 점과 교통량이 임의로 설정되었다는 것을 생각했을 때 알고리즘의 안정성을 보장하는 데이터로 보기 어려울 수 있다는 한계를 지닌다.

그러나 표 5에서 개선된 알고리즘의 결과에서 최소 근접 임계 값의 변화가 충돌 방지에 영향을 미치는 요소로 작용할 수

있다고 생각할 수 있다. 적절한 임계 값을 결정하는 것을 향후 연구 과제로 남길 수 있을 것이다.

그림 5는 각 기상 환경별로 교통량이 주어진 상황 속에서 지정된 맵을 주행하는 차량의 단계별 사진이다. 상황에 따라 시작 지점에서 충돌이 생기거나, 보행자 및 상대 차량과 충돌하는 경우, 주행 도중 정지하는 경우, 도착 지점을 인식하지 못한 채 도착 오류에 빠지는 경우, 인근 빌딩에 주행하는 경우 등의 예상하지 못한 상황도 실험 도중 마주하게 되었다. 이를 토대로 본 연구에서는 구현하지 못했던 시뮬레이션 기록 시스템의 필요성을 인지하였다. 또한, 향후 다양한 교통 상황을 다루는 주행 시나리오를 통해 개선된 충돌 회피 알고리즘을 구현할 수 있을 것이라고 고찰한다.

표 4. 주행 시간에 관한 결과

Table 4. Results of the driving time

	Previous algorithm	Improved algorithm
C	337 s	338 s
CN	400 s	386 s
MR	406 s	397 s
WC	407s	390 s
C x T	458 s	503 s
CN x T	612 s	650 s
MR x T	397 s	522 s
WC x T	474 s	420 s

표 5. 충돌 횟수에 관한 결과

Table 5. Results of the collision counts

	Previous algorithm	Improved algorithm
C x T	3	0
CN x T	1	0
MR x T	1	0
WC x T	0	0

## V. 결 론

본 논문에서는 외부 환경 요소를 고려한 자율주행 차량의 충돌 회피 알고리즘 및 시뮬레이션 개발의 필요성에 대해 인식하고 해당 내용을 구현하였다.

충돌 회피 알고리즘의 개선을 위해 기상 조건에 따른 가시거리의 변화를 가정하였고, 이에 따라 최소 근접 거리에 변화를 주도록 설정하였다. 또한, 시뮬레이션 구현에서는 4가지 기상 환경을 설정할 수 있도록 하였다. 구현을 위해 Carla 시뮬레이터를 사용하였고, 구현한 시뮬레이션 환경 속에서 기존 충돌 회피 알고리즘과 연구를 통해 개선한 알고리즘을 구동하여 결과를 비교하였다. 연구결과의 한계점이 존재하지만, 본 논문의 실험 결과 값들은 외부 환경 요소를 고려한 고도화된 충돌 회피 알고리즘 및 시뮬레이터 개발에 관한 향후 연구에 긍정적인 영향을 기대할 수 있을 것이다.

## Acknowledgments

본 과제(결과물)는 2022년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다. (2021RIS-004)

## References

[1] H. Thadeshwar, V. Shah, M. Jain, R. Chaudhari and V. Badgajar, "Artificial Intelligence based Self-Driving Car," *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, India, pp. 1-5, 2022.

[2] H. K. Kim and K. S. Heo, "Trends and Prospects of Self-driving Technology Research," *THE JOURNAL OF KOREAN INSTITUTE OF COMMUNICATIONS AND INFORMATION SCIENCES (J-KICS)*, Vol. 35, No. 5, June, 2018.

[3] R. Emuna, B. Avinoam, and B. Armin. "Deep reinforcement

learning for human-like driving policies in collision avoidance tasks of self-driving cars," arXiv preprint arXiv:2006.04218, Jun. 2020.

[4] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter and W. Stork, "Weather Influence and Classification with Automotive Lidar Sensors," *2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, France, pp. 1527-1534, 2019.

[5] S. H. Park, Y. W. Yun, H. G. Ko, H. R. Jeong and I. S. Yun, "Analysis of the Influence of Road-Traffic Conditions and Weather on the Take-over of a Conditional Autonomous Vehicle," *The Journal of The Korea Institute of Intelligent Transport Systems*, Vol. 19, No. 6, pp. 235-249, Dec. 2020.

[6] Y. N. Lee and G. W. Kim, "A study on the relationships between the fatal traffic accidents and meteorological factors," *Korean Police Studies Review*, Vol. 6, No. 3, pp. 125-164, Dec. 2007.

[7] J. Wei, J. M. Snider, T. Gu, J. M. Dolan and B. Litkouhi, "A behavioral planning framework for autonomous driving," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Dearborn, MI, USA, pp. 458-464, 2014.

[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, "CARLA: An Open Urban Driving Simulator," arXiv preprint arXiv:1711.03938v1, November. 2017.

[9] J. H. So, "Autonomous driving algorithm," *TTA Journal*, Vol. 197, pp. 63-69, Sep-Oct. 2021.

[10] E. Yurtsever, J. Lambert, A. Carballo and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, Vol. 8, pp. 58443-58469, March, 2020.

[11] M. Jaswanth, N. K. L. Narayana, S. Rahul, and M. Supriya. "Autonomous Car Controller using Behaviour Planning based on Finite State Machine," *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, pp. 296-302, 2022.

[12] B. Parden, M. Cap, S. Z. Young, D. Yershov and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Transactions on Intelligent Vehicles*, Vol. 1, No. 1, pp. 33-55, March. 2016.

[13] N. D. Van, M. Sualeh, D. Kim and G-W. Kim, "A Hierarchical Control System for Autonomous Driving towards Urban Challenges," *Applied Sciences*, Vol 10, No. 10: 3543, May. 2020.

[14] H. C. Lee, Human-Vehicle Interaction in Conditional Autonomous Vehicle: Take-over Requests in Environments, Ph.D. dissertation, Hanyang University, 2020.

[15] S. Li, B. Phil, G. Weihong and N. Anil, "Investigation of

older driver's takeover performance in highly automated vehicles in adverse weather conditions," *IET Intelligent Transport Systems*, Vol. 12, No. 9, pp. 1157-1165, Jul. 2018.

[16] Carla-simulator CARLA. 2021. carla. San Francisco (CA): GitHub; [accessed 2023 Feb 18]. Available: <https://github.com/carla-simulator/carla/tree/0.9.13/Python/API/carla/agents/navigation/>.



**이 서 영** (Seo-Young Lee)

2021년 8월 : 한국기술교육대학교 컴퓨터공학과 졸업  
2022년 3월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정  
※관심분야 : AI, 웹, 컴퓨터 비전 등



**용 성 중** (Sung-Jung Yong)

2020년 8월 : 한국기술교육대학교 대학원 컴퓨터공학과 공학석사  
2021년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 박사과정  
※관심분야 : AI, 빅데이터, 추천 시스템, 웹 등



**박 호 경** (Hyo-Gyeong Park)

2021년 8월 : 한국기술교육대학교 컴퓨터공학 학사  
2021년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정  
※관심분야 : AI, 빅데이터, 추천 시스템 등



**유 연 휘** (Yeon-Hwi You)

2022년 2월 : 한국기술교육대학교 컴퓨터공학과 공학사  
2022년 3월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정  
※관심분야 : AI, 빅데이터, 추천 시스템 등



**문 일 영** (Il-Young Moon)

2005년 2월 : 한국항공대학교 항공통신정보공학과 공학박사  
2005년 3월 ~ 현재 : 한국기술교육대학교 컴퓨터공학과 교수  
※관심분야 : AI, 무선인터넷 응용, 무선 인터넷, 모바일IP 등