

# 교통 정보 전달을 위한 ASAM 표준 기반의 지도 서버와 공간 데이터베이스

김 인 호<sup>1)</sup>

한양대학교 미래자동차공학과<sup>1))</sup>

## Map Server and Spatial Database based on ASAM Standard for delivering Transport Information

Inho Kim<sup>1)</sup>

<sup>1))</sup>Department of Automotive Engineering, Hanyang University, Seoul 04763, Korea

**Abstract** : The goal of this project was to establish a map server and a spatial database based on a high definition map that can be used by autonomous driving. Therefore, the OpenDRIVE map format was analyzed in detail, and a software architecture consisting of two clients and a server was devised. Next, while actually developing the front-end and the back-end, not only GUI that can be easily used by users, but also inclusion calculation logic according to the shape of road, and a fast and intuitive database storage structure were completed. After that, through tests in which actual files were inserted, accurate storage of the program and precise location calculation were verified. This project is of great significance in that it presents the concept of how to develop a map server in an cooperative automated driving system.

**Key words** : Map Server(지도 서버), Spatial Database(공간 데이터베이스), OpenDRIVE(오픈드라이브), Collector(수집자), User(이용자), Cooperative Automated Driving(자율협력주행)

### 1. 서 론

차량 클라이언트와 연결된 지도 서버와 공간 데이터베이스는, 운전자의 쉽고 효율적인 이동에 있어 필수적인 프로그램이 되었다. 도로의 수많은 교통 데이터를 주고받음으로써 더욱 효율적이고 안전한 주행을 선택할 수 있기 때문에, 이러한 교환에 있어 중심적인 역할을 하는 서버의 중요성이 계속 증가하고 있다. 따라서 현재 많은 모빌리티 및 IT 기업들이, 자체적인 지도 서버 구축과 관련 서비스 개발을 적극적으로 진행하고 있다. 현대자동차의 경우, 순정 내비게이션을 서버에 연결하여, 차량이 실시간으로 도로 정보를 확인하고 최적의 경로를 찾을 수 있도록 유도한다.<sup>1)</sup> 카카오모빌리티 역시, 지도 서버와 연동된 MaaS 플랫폼을 개발 및 운영함으로써, 카셰어링, 카세어링, 마이크로모빌리티 등의 다양한 이동 서비스를 제공하고 있다.<sup>2)</sup>

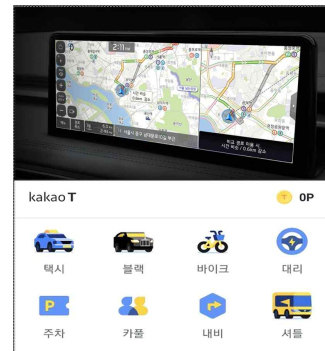


Photo. 1 Hyundai Motor's navigation and Kakao Mobility's maas platform

지도 서버와 공간 데이터베이스의 작동 과정을 Fig 1에서 간략히 나타내 보았다. 어떤 도로 데이터가 들어오면, 서버는 해당 데이터가 무슨 정보를 담고 있는지 파악한다. 보통 도로 모양, 교차로, 규정 속도와 같은 정적 정보와, 신호등, 통행량, 돌발 상황과 같은 동적 정보로 이를 나눌 수 있으며, 공간 데이터베이스에서 해당 정보가 어디

에 위치하는지 확인하고 저장을 진행한다. 이후 유저로부터 요청이 오면, 지도 서버는 요청에 맞는 데이터를 판단하고, 공간 데이터베이스에서 해당 정보를 꺼내와 유저에게 보내주는 방식으로 데이터 전송을 완료한다. 즉 간단히 말해, 공간 데이터베이스는 교통 정보를 담는 저장소 역할을 하며, 지도 서버는 클라이언트와 데이터베이스 사이의 원활한 중재를 담당한다.



Fig. 1 Operation process of map server and spatial database

최근에는 자율협력주행에 대한 연구가 지속적  
으로 이뤄짐에 따라, 자율주행용 정밀지도를 기반  
으로한 지도 서버 구축이 중요한 주제로 떠오르고  
있다. 자율협력주행이란, 자체 센서로 인지한 정  
보뿐만 아니라, V2X 통신을 통해 얻은 도로 환경  
정보들도 자율주행에 활용하는 기술이다.<sup>3)</sup> 자율  
주행 Level4를 달성하기 위해선, 이러한 자율협  
력주행 기술이 필요하다는 의견이 지배적이며, 특  
히 자율주행이 이용할 수 있는 세밀한 수준의 데  
이터를 제공하는 것이 중요하다고 보고 있다.<sup>4)</sup> 즉  
기존보다 더욱 정밀한 수준의 지도 서버를 바탕으  
로 교통 정보를 주고받을 필요가 있다는 것이다.



Photo. 2 Cooperative automated driving

따라서 이번 연구에서는, 자율주행용 정밀지도  
를 기반으로 도로 정보를 주고받을 수 있는 서비  
스 프로그램을 개발해 보았다. 해당 지도 포맷으  
로, 많은 기관들에서 표준으로 사용하는 ASAM  
의 OpenDRIVE를 선정하였다.<sup>5)</sup> 차량의 경우는,  
웹 환경에서의 가상 클라이언트를 가정함으로써,  
서버 및 데이터베이스와의 전체적인 데이터 교환  
시나리오 구현에 중점을 맞추기로 결정하였다.

## 2. OpenDRIVE

ASAM은 자동차 산업에 대한 소프트웨어 아키텍처와 측정 기준을 제정하는 비영리 기구로, 많은 기업 및 기관에서 이 표준을 토대로 모빌리티 시스템을 개발하고 있다.<sup>6)</sup> 자율주행에 관련된 개방형 지도 포맷 역시 제정되어 있는데, 그것이 바로 이번 연구에서 적용한 OpenDRIVE이다. 이는 3차원 좌표계를 토대로 도로, 레인, 오브젝트, 커넥션 등의 도로 정보들을 XML 형식으로 담고 있다. 특히 벡터 및 다항 함수를 이용하여 기하학적 형태를 세밀하게 표현하기 때문에, 실제 공간의 정확한 정보들을 잘 반영할 수 있다.<sup>7)</sup>

```
XML Read 99
<doc name="Road 39" length="1.379521227443047e+2" id="39" junction">Y
  <id>
    <predecessor elementType="junction" elementId="669" />
    <successor elementType="junction" elementId="720" />
  </id>
  <type>s:0.0000000000000000e+0 type="boun"
    <speed>max'55' units="mph"/>
  </type>
  <curveInfo>
    <geometry>s:0.0000000000000000e+0 w:6.82954618210261622e+1 x:2.112110599012624e+1 h:1.137532270235350e+0 length=4.438808246099741e+1
      <id>1
    </id>
    <geometry>s:4.438808246099676e+1 w:2.281499080178717e+1 x:2.19300807934570313e+1 h:1.137532270235350e+0 length=7.520557054612745e+1
      <id>2
    </id>
    <geometry>s:7.520557054612745e+1 w:4.40915448648793e+1 x:2.632765532410796e+1 h:1.137532270235350e+0 length=1.5597960160779396e+0
      <curve>curvature=2.000000000000000e+0
    </curve>
    <geometry>s:1.189403004792146e+2 w:5.0468985218962395e+1 x:2.1629599282103275e+1 h:1.402518623437001e+1 length=1.897887779512555e+1
```

Fig. 2 XML format OpenDRIVE

## 2.1 좌표계

지도의 기초적인 토대가 되는 좌표계에 대해, OpenDRIVE는 총 3가지의 종류를 사용한다. 첫 번째는  $x/y/z$  축으로 이루어진 관성 좌표계로, 고정된 원점에서 절대 좌표를 형성한다. 즉 모든 요소들은 이 좌표계를 기준으로 공간상의 명확한 위치를 결정할 수 있다. 두 번째는  $s/t/h$  축으로 이루어진 도로 기준 라인 좌표계이다. 도로의 기준선 방향을  $s$ 축으로 잡고, 그에 대한 평면 수직 방향이  $t$ , 그리고 위로 나가는 방향이  $h$ 가 된다. 이 좌표계를 이용하여 도로선의 형태를 결정할 수 있는데, 형태의 시작점을 원점으로 잡고, line, arc 등의 모양을 결정한 다음, 세부적인 곡률이나 함수의 계수를 같이 적어줌으로써 작성할 수 있다. 마지막으로  $u/v/z$  축으로 이루어진 지역 좌표계는, 도로 안의 레인이나 오브젝트와 같은 요소들을 기준으로 하는 좌표계이다. 이 지역 좌표계와 도로 기준 라인 좌표계는, 축을 기준으로 시계방향으로의 roll/pitch/heading 회전이 가능하다. 이러한 회전을 통해 요소들의 종방향, 횡방향 기울기를 나타낼 수 있어, 더욱 정밀한 표현을 할 수 있다.

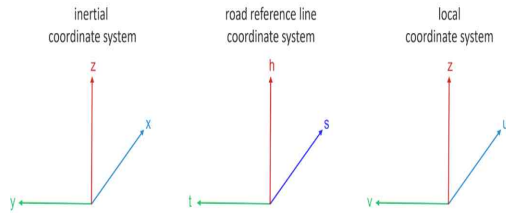


Fig. 3 Coordinate system in OpenDRIVE

## 2.2 로드

로드는 여러 레인을 포함하는 도로상의 기본적인 요소이다. 처음에는 로드의 이름과 길이 및 아 이디 정보를 보여주며, 다음은 링크 구역을 통해 앞뒤로 이어진 다른 로드와의 관계를 표현해 준다. 좌표계 시스템에서 언급했던 도로 기준 라인은 planview 항목에서 정의가 되며, 라인의 모양이 달라질 때마다 새로운 s/t/h 지오메트리를 추가한다. 이후 도로의 고도 및 기울기 정보를 3차 함수로 표현하는 elevationProfile 필드가 있으며, 이를 통해 3차원 공간 표현이 가능하다. lanes 파트부터는 도로 안에 있는 레인들 정보들이 상세하게 정의가 되어 있다.

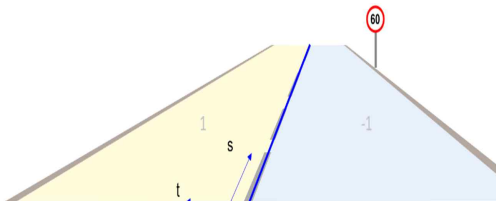


Fig. 4 Road in OpenDRIVE

## 2.3 레인

도로 안에 있는 여러 레인은, 도로 기준 라인을 id 0의 중앙 라인으로 판단하여, t축 좌표가 높아지는 방향으로 1씩 올라가는 id가 부여되며, 반대로 t가 낮아지는 방향으로 1씩 작아지는 id가 부여된다. 즉 중앙라인을 기준으로 t값이 높은 레인은 s축 방향과 반대되는 방향의 진행이 default로 설정되며, 낮은 레인은 그 반대와 같다. 이 외에도 레인별 너비, 제한 속도, 타입 등의 다양한 정보 역시 지정해 줄 수 있다.

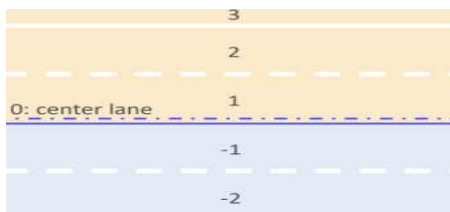


Fig. 5 Lane in OpenDRIVE

## 3. Architecture

위와 같은 OpenDRIVE 포맷을 기반으로 한 교통 정보 교환 시스템의 전체적인 아키텍처는 Fig 6과 같다. 먼저 클라이언트는 도로 정보를 수집하는 Collector와 해당 정보를 요청하는 User로 나뉘게 되며, 사용자가 처음으로 접근하는 페이지인 Home에서 둘 중 하나를 결정할 수 있다. 이후 Collector와 User에서 각각에 맞게 메시지를 보내면, 해당하는 Controller 모듈부터 처리 로직이 실행된다. Dao 모듈로 그 흐름이 넘어가면, 이곳에서 데이터베이스와의 소통이 진행되고, 적절한 저장 위치에서 교통 데이터를 업로드하거나 다운로드하는 동작이 일어난다. 이후에는 다시 역과정을 거쳐서, 동작 결과를 바탕으로 응답 내용을 작성한 뒤, 최종적으로 클라이언트에게 해당 메시지를 보내게 된다.

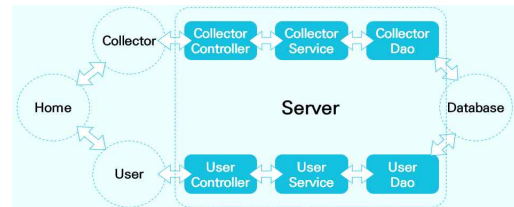


Fig. 6 Software architecture

## 4. Front-end

프론트엔드는 JSP 포맷 바탕의 HTML 언어로 작성하였다. 아키텍처에서 언급한 것처럼 총 3가지의 페이지가 존재하며, 이는 Home, Collector, User로 나뉜다.

### 4.1 Home

사용자가 웹상에서 프로그램에 접속하면, 처음으로는 Home이라는 페이지로 진입한다. 해당 페이지의 경우, Collector 버튼과 User 버튼을 통하여 각각의 클라이언트로 쉽게 넘어갈 수 있도록 제작하였다.

### 4.2 Collector

Collector는 새로운 OpenDRIVE 파일이나 특정 위치의 동적 상황을 서버에 업데이트하는 역할을 한다. Photo 3을 보면, 맨 왼쪽의 파일 선택 버튼에서 원하는 OpenDRIVE 파일을 선택할 수 있다. 중앙의 두 입력창에서는, 각각 경위도 좌표와

특정 상황을 입력할 수 있다. 이때 사용자가 편리하게 좌표를 입력할 수 있도록, 구글 맵스 API를 연동하였다. 즉 지도상에서 특정 위치를 클릭함으로써 경위도 좌표를 자동으로 입력할 수 있다. 마지막으로 가장 오른쪽에 있는 Send 버튼을 눌러 입력된 정보들을 서버에 전송한다.

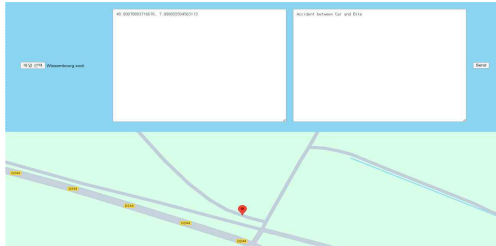


Photo. 3 Frontend of collector client

### 4.3 User

User는 Collector와 반대로, 특정 위치에 대한 OpenDRIVE 파일과 도로 정보를 서버로부터 받아 출력하는 역할을 한다. 따라서, Coordinate 입력창에 원하는 경위도 좌표를 작성하는 것이 우선이다. 여기서도 구글 맵스 API를 연동하여, 쉽게 위치를 입력할 수 있도록 제작하였다. 이후 오른쪽의 Receive 버튼을 누르면, 서버로부터 해당 좌표를 포함한 도로의 정보를 가져와 Situation 박스에 출력한다. 만약 왼쪽의 선택 리스트에서 No 항목을 Yes로 바꾼 뒤 Receive를 누르면, 도로 정보 대신 해당 좌표를 포함하는 OpenDRIVE 파일 자체를 다운로드받게 된다.

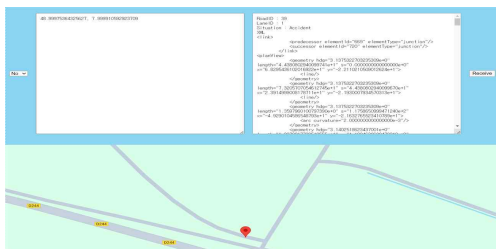


Photo. 4 Frontend of user client

## 5. Back-end

백엔드는 서버와 데이터베이스로 구성된다. 서버는 Spring 프레임워크의 MVC 모델을 적용하여 Java 언어로 작성하였다. 데이터베이스로는 MariaDB를 이용하며, 서버의 jdbcTemplate 객체를 통해 이를 조작하는 SQL문을 작성한다.

### 5.1 Server

서버에서는 Collector의 요청과 User의 요청을 처리하는 부분이 각각 따로 존재한다. 전자의 경우, Collector가 보낸 OpenDRIVE 파일을 저장하는 로직을 우선적으로 실행한다. 이때 원본을 안정적으로 유지하기 위해, 로컬 저장소에 파일을 저장한 뒤 내용을 분석하여 데이터베이스에 구조화한다. 이후 동적 상황 입력이 있는 경우, 해당 좌표가 어느 파일의 어떤 도로의 몇번 레인에 있는지 파악하여 데이터베이스에 정보를 저장한다.

다음은 경위도 좌표의 도로 위치를 파악하는 기하학적 계산이다. 먼저 어느 OpenDRIVE에 좌표가 포함되는지를 파악하기 위하여, 해당 파일에 적힌 원점 좌표와 동서남북 범위를 통해 포함 관계를 확인한다. 이때 도단위로 넘어온 입력 경위도가 원점에 맞는 미터단위 좌표로 바뀌게 된다.

이후 해당 파일에서 어느 도로 및 레인에 좌표가 위치하는지 파악해야 한다. 먼저 사전적으로 알고 있는 정보들은 다음과 같다.

- 1) 도로기준선 원점 :  $sx, sy(m, m)$
- 2) 도로기준선 각도 :  $sh(rad)$
- 3) 도로 길이 :  $len(m)$
- 4) 도로 곡률 :  $cur(1/m)$
- 5) 레인들의 폭 :  $width(m)$
- 6) 입력좌표 :  $x, y(m, m)$

이때 곡률이 0인가 아닌가에 따라 도로의 모양이 Line 또는 Arc로 분류되고, 이에 따라 계산 과정도 달라진다. 먼저 Line의 경우는, 입력좌표와 도로기준선이 이루는 각도( $a$ )를 구했을 때, 해당 각도가  $\pm 90$ 도 내에 존재하는지 확인한다. 또 입력좌표를 도로기준선으로 내렸을 때의 길이( $b$ )가,  $len$ 보다 작은지 확인한다. 마지막으로, 입력좌표와 도로기준선까지의 거리( $c$ )가  $width$  범위 안이면, 해당 레인에 좌표가 존재한다고 판단한다.

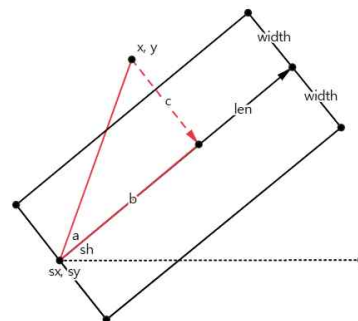


Fig. 7 Geometric calculation to check inclusion relationship in line-shape road

Arc의 경우는 원형적 곡선을 고려해야 한다. 따라서 먼저 도로기준선으로부터 수직으로 반지름만큼 떨어져 있는 원의 중심좌표  $cx, cy$  (m, m)를 찾는다. 다음으로는 호의 길이인  $len$ 를 이용하여, 호의 각도 범위( $range$ )를 도출한다. 이후 중심좌표와 입력좌표의 각도를 구하고, 해당 각도가 호의 각도 범위 안에 들어가는지 확인한다. 마지막으로, 중심좌표와 입력좌표의 거리를 구하여, 해당 거리가  $width$  범위 안이면, 입력좌표가 해당 레인에 존재한다고 판단한다.

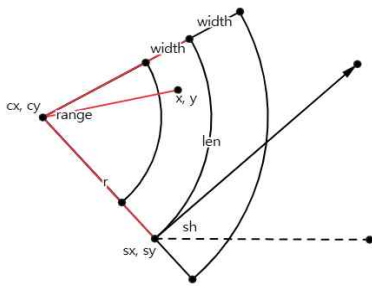


Fig. 8 Geometric calculation to check inclusion relationship in arc-shape road

서버에서 User의 요청을 처리하는 부분 역시, 위와 같은 기하학적 계산을 통해 입력 좌표를 포함하는 OpenDRIVE 파일과 레인을 찾는다. 이후 해당 도로와 레인의 정보를 형식에 맞게 가공하여 유저에게 다시 전송한다. 파일 역시 유저가 다운로드할 수 있도록 response 메시지를 통해 직접적으로 전달한다.

## 5.2 Database

데이터베이스의 구조는 다음과 같다. 먼저 파일의 메타데이터를 담는 Header Table이 존재한다. 새로운 OpenDRIVE가 들어올 때마다, 해당 지도의 중심 경위도와 동서남북 범위가 위 테이블에 저장된다. 즉 특정 경위도 좌표가 입력으로 들어왔을 때, 파일마다 일일이 들여다볼 필요 없이, Header Table 안에서의 빠른 비교를 통해, 해당 좌표를 포함하는 지도를 쉽게 찾을 수 있다. 이어서, OpenDRIVE마다 할당된 Road Table이 존재한다. 각 행마다 지도 속에 존재하는 하나의 도로에 대한 정보를 담고 있으며, 구체적으로 도로기준선, 레인, 링크, 상황 등의 데이터를 포함하고 있다. 위와 같은 저장 구조를 바탕으로, 원하는 데

이터를 직관적이고 빠르게 다룰 수 있다.

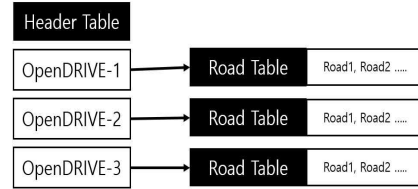


Fig. 9 Database structure

## 6. Implementation

개발된 프로그램을 고정 IP 및 Port 주소로 접근할 수 있도록, AWS EC2 클라우드에 해당 서버와 데이터베이스를 등록하였다. 따라서 아래 주소로 접속하여 프로그램을 이용할 수 있다.

<http://43.203.236.58:8080/map/>

지도 내용을 정확하게 저장하고 입력 좌표의 정밀한 위치를 찾을 수 있는지 확인하기 위해, 여러 OpenDRIVE 파일들을 이용하여 실제 테스트를 진행하였다. 우선 Collector 관점의 테스트 결과, 입력한 파일이 데이터베이스에 설계한 구조대로 테이블화 되는 것을 확인할 수 있었다. 또 User 관점의 테스트에서는, 도로 위의 좌표들을 수작업으로 계산하여 뽑아낸 뒤, 해당 좌표를 입력 좌표로 대입하여 결과를 비교함으로써, 프로그램의 정확한 레인 파악 능력을 확인하였다.

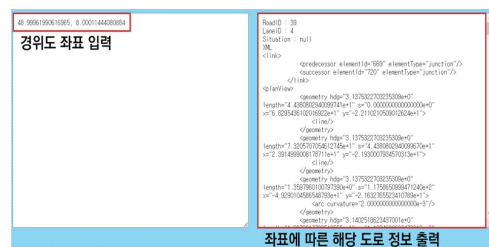


Photo. 5 Test result in user

추가적으로 OpenDRIVE를 3차원적으로 보고 싶은 경우, 해당 파일을 RoadRunner에 Import하여 기하학적 형태를 자유롭게 관찰할 수 있다.<sup>8)</sup>

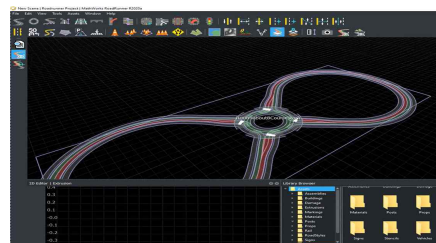


Photo. 6 OpenDRIVE in RoadRunner



## 7. 결 론

이번 프로젝트의 목표는, 자율주행이 이용할 수 있는, 정밀지도 기반의 지도 서버와 공간 데이터베이스를 구축하는 것이었다. 따라서 OpenDRIVE 지도 형식을 구체적으로 분석하고, 두 클라이언트와 서버 단으로 이루어진 소프트웨어 아키텍처를 구상하였다. 다음으로 실제로 프론트엔드와 백엔드를 개발하면서, 사용자가 쉽게 이용할 수 있는 GUI는 물론, 도로의 형태에 따른 포함관계 계산 로직과, 빠르고 직관적인 데이터베이스 저장 구조까지 완성하였다. 이후, 해당 서버와 데이터베이스를 AWS에 올렸으며, 실제 파일들을 대입한 테스트들을 통해, 프로그램의 정확한 저장과 정밀한 위치 계산을 검증하였다.

이번 프로젝트는, 자율협력주행 시스템에서의 지도 서버를 어떻게 개발해야 할지에 대한 개념을 던져준다는 점에서 큰 의미가 있다. 특히 현재 표준으로 사용되는 OpenDRIVE를 적용하였기 때문에, 구현한 데이터 교환 시나리오와 위치 계산 로직이 실제 시스템에서도 충분히 적용될 수 있을 것이라 판단된다. 따라서 해당 연구를 기반으로, 실제 도로 환경과 연결된 더 많은 연구가 이루어지길 기대한다.<sup>9)</sup>



Photo. 7 Cooperative automated driving in real world

## 후 기

약 1년 동안 프로젝트를 진행하면서, 평소에 궁금하던 자율주행 정밀지도를 공부하고, 이를 Spring 프레임워크에 직접 적용하는 값진 경험을 할 수 있었다. 특히 지도 데이터를 처리하는 로직을 스스로 설계하고 구현함으로써, 해당 도메인과 관련된 개발 역량을 크게 향상할 수 있었다. 연구를 진행하며, 한양대학교 미래자동차공학과 구성원들에게 많은 질문을 구할 수 있어 감사했고, 특히 전체적인 개발 방향과 여러 가지 OpenDRIVE 관련 도구들을 소개해주신 조한신 교수님께 큰 감사를 드린다.

## References

- 1) Hyundai-Motor, The Connected Car Technology applied to Hyundai Car, <https://brunch.co.kr/@b6e2969fe95743c/178>, 2022.
- 2) Mosangsil-Team, How Kakao Mobility Company Makes Money by Platform, <https://maily.so/mosangsil.newsletter/posts/9ba986>, 2021.
- 3) S. Kim, D. Kim, S. Lee, J. Shin, S. Cho, B. Moon and K. Park, "A Study on the Development Methodology of Functional Safety Evaluation Scenario for V2V Communication of Urban Cooperative Autonomous Driving System," Transactions of KSAE, Vol.29, No.5, pp.437-449, 2021.
- 4) D. Kim, S. Kim, S. Lee and K. Park, "Development of Fail-Safety Evaluation Environment for V2I-Based Cooperative Autonomous Driving System," KSAE Annual Conference Proceedings, pp.646-646, 2019.
- 5) Global-Auto-News, Trends in Standardization of Autonomous Driving, <https://post.naver.com/viewer/postView.naver?volumeNo=31692141&memberNo=32414000&searchKeyword=%EC%9E%90%EC%9C%A8%EC%A3%BC%ED%96%89&searchRank=49>, 2021.
- 6) Wikipedia, Association for Standardisation of Automation and Measuring Systems, [https://en.wikipedia.org/wiki/Association\\_for\\_Standardisation\\_of\\_Automation\\_and\\_Measuring\\_Systems](https://en.wikipedia.org/wiki/Association_for_Standardisation_of_Automation_and_Measuring_Systems), 2023.
- 7) ASAM, ASAM OpenDRIVE Specification, [https://publications.pages.asam.net/standards/ASAM\\_OpenDRIVE/ASAM\\_OpenDRIVE\\_Specification/latest/specification/index.html](https://publications.pages.asam.net/standards/ASAM_OpenDRIVE/ASAM_OpenDRIVE_Specification/latest/specification/index.html), 2024.
- 8) Mathworks, Import and Export the OpenDRIVE Map Foramt with RoadRunner, <https://kr.mathworks.com/videos/importing-and-exporting-scenes-with-roadrunner-1597837006663.html>, 2020.
- 9) J. Lee, Beyond Autonomous Driving to the Era of Cooperative Automated Driving, [https://gnews.gg.go.kr/news/news\\_detail.do?number=202205241628085231C048&s\\_code=C052](https://gnews.gg.go.kr/news/news_detail.do?number=202205241628085231C048&s_code=C052), 2022.