

# IoT와 메타버스를 이용한 스마트 항만 교통제어 시스템 제안 : 자율주행차를 위한 스마트 신호등

오유나, 신이서, 전예린, 한예송  
이화여자대학교 전자전기공학과

una0409@ewhain.net, s990121@ewhain.net, apocalypserisen@ewhain.net, jately0819@ewhain.net

## Proposal for Smart Port Traffic Control System Using IoT and Metaverse : Smart Traffic Lights for Self-driving Yard Tractors

Yuna Oh, Yiseo Shin, Yerin Jeon, Yea Song Han

Dept. of Electronic and Electrical Engineering, Ewha Womans University

### 요 약

본 논문은 항만 완전 자동화를 위하여, 자율주행 트랙터와 스마트 신호등을 도입한 IoT 기반 스마트 항만 교통제어 시스템을 메타버스를 통해 제안한다.

### 1. 서론

우리나라의 스마트 항만은 현재 완전 자동화가 이루어지지 않았으며, 다른 나라의 구조를 도입하기 어려운 상태이다. 타국의 항만 완전 자동화 방식 또한 반영구적인 구조이며 센서에 의존한다는 단점이 존재한다. 이에 본 연구는 기존의 완전 자동화에 주로 사용되었던 센서의 의존도를 낮추고, 이미지 프로세싱과 통신을 이용하는 새로운 스마트 항만 시스템을 제안한다.

### 2. 최적 경로 선정 알고리즘

최적 경로 문제는 네트워크 이론에서 가장 기본적이고 중요한 문제 중의 하나이며, 최근에 연구가 활발한 지능형 교통체계(Intelligent Transportation System : ITS) 분야 중 첨단 여행자 정보체계(Advanced Traveler Information System : ATIS)에서 효율적인 경로 정보를 제공하기 위하여 필수적인 기법이다[1]. 일반적으로 최적 경로를 탐색하는 알고리즘은 통행시간, 통행비용 또는 경로 길이 등 하나의 속성만을 최우선으로 고려하며, 통행자는 자신의 목적에 맞는 경로를 최적 경로로 선택한다. 그러나 실제로 통행자가 단일 속성만을 고려하여 경로를 선택하는 경우는 드물며, 통행시간이나 경로 길이 또는 통행자의 개인적인 선호 등과 같은 다양한 속성들을 종합적으로 고려하여 선택한다.

그러므로 최적 경로 탐색 과정에서 이들 속성을 복합적으로 고려해야 할 것이다. 기존 알고리즘으로는 Dijkstra, Moore 등 다양한 기법이 개발되었으며, 다중 기준을 고려한 알고리즘으로는 파레토 최적을

이용한 “A bicriterion shortest path algorithm”(Climaco, Martins, 1982)과 “On Finding dissimilar paths”(V.Akgun, E. Erkut,2000), “On finding dissimilar Pareto-optimal paths”(Paolo Dell’Olmo, 2004)등이 있다[2][3].

### 3. 제안 방법

본 논문은, 실시간으로 차량의 교통량을 파악하고, 최적의 경로를 탐색하여 자율주행 트랙터에 명령을 전송하는 스마트 통신 신호등을 이용한 스마트 항만 교통제어 시스템을 제안한다.

차량의 교통량을 파악하기 위해서 차량의 데이터를 수집하고, 도로의 교통상황을 객체 검출을 통해 수집한 후 수집된 데이터를 기반으로 최적 선정 알고리즘을 구동하여 최적의 경로를 도출한다. 도출한 경로에 대한 신호를 차량에 전달하여 차량은 최적의 경로를 위한 현재 교차로에 대한 신호를 수신한다.

#### 3.1. 차량 교통량 파악 및 차량 데이터 수집

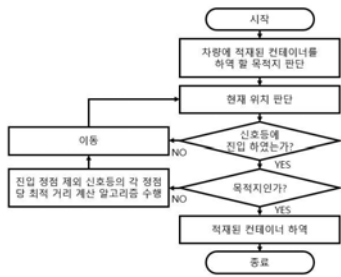
도로 상황 및 운행 차량 상태를 파악하기 위해서 신호등에서는 카메라를 활용한다. 본 연구 사용 기술은 YOLO인데, 우선 실제 도로와 달리 통제 가능한 상황의 데이터를 처리하기 때문에 부하를 줄이기 위해 동영상을 분절해 이미지로 저장하는 작업이 필요하다. 또한, 검출에 필요한 ROI(Region of Interest)를 지정해 비관심 영역을 무시함으로써 측정 속도를 개선한다. 신호등은 단순 도로 상황 파악이 아닌 차량에의 지시 역할까지 겸해야 하는데, 지시를 받을 차량을 특정하고 검증하기 위해서는 두

가지 정보가 필요하다. 첫째, 신호등이 수신한 영상에서 검출해 낸 도로 상황 및 차량 정보이다. 둘째, 차량에서 최초 지시한 상하역 정보에 따라 차량별로 사전 배정받은 목적지까지의 최적 경로이다. 이 두 가지 정보를 조합해 차량이 해당 구간에서 어떤 경로로 향해야 할지 최종적으로 결정한다.

**3.2. 최적 경로 선정 알고리즘**

Dijkstra 알고리즘은 출발 Node에서 시작하여 각 Node에 반복해서 적은 비용으로 도달할 수 있는 값을 업데이트하는 방식으로 목적 Node까지의 최소 비용을 계산할 수 있는 알고리즘이다.[4]

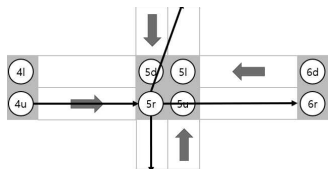
제안하는 최적 경로 선정 알고리즘은 Dijkstra's 알고리즘을 기반으로 여러 경로 중 최적화된 경로를 선택한다. 또한, 특정 경로에서의 거리 비용이 증가한 경우 최적 경로가 변경되어야 한다.[5] 경로 선정을 위한 전체 알고리즘은 (그림 1)과 같다. 본 연구에서 구현한 알고리즘은 차량이 도착한 각 신호등에서 목적지까지의 거리 비용을 계산하여 가장 비용이 적게 드는 방향으로 이동하는 부분 최적화를 신호등마다 실행하여 최적 경로를 선정한다.



(그림 1) 최적 경로 선정 알고리즘

**3.2.1. 전체 경로**

먼저 최적 경로를 선정하기 위해서는 전체 경로에 대한 데이터가 필요하다. 항만에서의 자율주행차량에 대한 최적 경로 탐색을 위한 연구이므로 한정된 공간 안에서의 최적 경로를 선정하기 때문에 사전에 모든 경로에 대한 데이터를 수집할 수 있다. (그림 2)와 같은 모양으로 경로 데이터를 수집한다.



(그림 2) 경로 수집

네모난 영역이 신호등이 관장하는 영역이고 해당 영역에서 차량이 진입할 수 있는 도로별로 4개의 Node를 생성한다. 각 Node에서 도달할 수 있는 모든 Node를 거리에 따른 Weight를 가진 Arc로 연결하여 Weighted Directed Graph를 그려낸다. 모든 도로는 차선별 일방통행의 형태이므로 Directed Graph로 그릴 수 있다. 이 과정을 모든 신호등에 대

해 반복하여 전체 경로 데이터를 수집한다.

$$w_{ij} = \alpha D_{ij} + \beta w_{t,ij}$$

Weight(w)는 제안된 i Node에서 j Node로의 Arc 당 거리(D)와 3.1. 과정에서 수집된 Arc에서의 교통량 데이터(w<sub>t</sub>)의 합으로 구성된다. 이 연구에서 사용된 파라미터는 거리 계수 α=1, 교통량 계수 β=1000이다. 따라서 Weight의 값은 교통량에 따라 실시간으로 변동한다.

**3.2.2. 차량 데이터 수집 및 현재 위치 판단**

알고리즘에서는 차량이 컨테이너를 하역할 목적 Node 데이터, 차량의 현재 위치 데이터가 필요하다. 신호등은 3.1.에서 신호를 받을 차량을 특정하여 서버에서 해당 차량에 대한 목적 Node 데이터를 수집하고, 차량은 해당 과정에서 업데이트되는 서버 데이터를 이용하여 현재 위치를 신호등에 기반을 둔 상대 위치로 파악한다. 수집된 데이터를 기반으로 현재 위치가 목적 Node인지 파악하여 다음 단계를 수행한다.

**3.2.3. 최적 거리 계산 알고리즘**

각 신호등당 최적 거리를 계산하는, 부분 최적화 방식으로 사전에 입력된 Weighted Directed Graph 형태의 전체 경로를 가지고 최적 거리를 계산한다. 최적 거리 계산을 위해 Dijkstra's 알고리즘을 사용하는데 그 이유는 특정 시작 Node와 목적 Node를 지정하여 거리 비용(Weight)을 가지고 해당 Node 간의 최단 거리 비용을 계산할 수 있기 때문이다.

최적 거리 계산 과정은 다음과 같다. 차량이 진입한 신호등의 4개 Node 중 차량 진입 Node를 추출한다. 해당 Node를 제외한 1~3개의 Node를 출발 Node로 지정하여 목적 Node까지의 거리 비용을 계산한다. 계산된 각 출발 Node 중 가장 거리 비용이 작은 방향으로 이동 신호를 전송한다.

**3.2.4. 이동 신호**

자율주행차량에 전달되는 이동 신호의 종류는 신호등의 Node 당 3개씩 총 12개의 신호로 구성되어 있다. 사람의 인지 능력에 의존하여 직진 및 좌회전만을 제공하는 기존 신호 시스템과 달리 세부적인 명령이 수행되어야 하므로 각 신호등에서의 신호를 세분화하였다. 신호등에 진입한 위치의 Node에서의 좌회전, 우회전, 직진 신호로 구성되어 있다.

**3.3. 차량 데이터 송신 및 차량 이동**

결정된 최적의 경로를 기반으로 차량이 이동해야 할 방향은 서버가 차량에 직접 제시한다. 서버는 경로 선정 후 차량의 이동 방향 정보를 SQL을 활용하여 데이터베이스에 업데이트한다. 모든 차량과 신호등은 실시간으로 데이터베이스를 업데이트하며, PHP의 POST 등의 메소드를 사용하여 수집한다. 스마트 신호등은 보행자의 안전을 위해 각 차량의 주행 재개 여부와 이동 방향을 LED 모듈을 통해 표시

하여야 하므로, 해당 정보를 도로 위 전시한다. 차량은 지시받은 방향으로 이동한다. 이때 주행 방향 정보를 서버에서 신호등으로, 신호등에서 차량으로 전달하지 않는 이유는 시간 단축 및 효율성을 위함이다.

**4. 실험 결과**

**4.1. 최적 경로 선정 알고리즘**

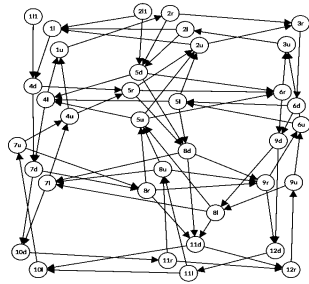
제안한 알고리즘을 검증하기 위해, 차량에 적재되는 컨테이너와 차량 교통량에 따라 변화하는 거리 비용을 기반으로 한 웹 기반 메타버스 시뮬레이션 페이지를 제작하여 확인하였다.

**4.1.1. 실험 환경 구성**

웹 기반 메타버스 시뮬레이션 페이지를 활용하여 최적 경로 선정 알고리즘을 실험하였다. 해당 환경에서 구현한 전체 지도는 (그림 3)과 같다.



(그림 3) 실험 지도



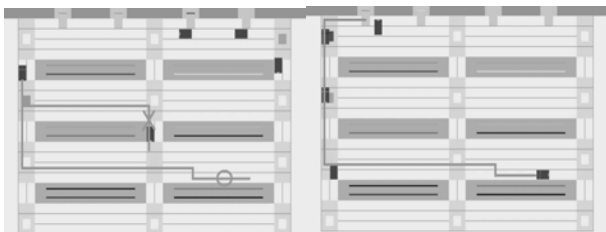
(그림 4) 생성된 Directed Graph

(그림 3)의 지도를 기반으로 (그림 4)의 Directed Graph를 생성할 수 있다. 생성된 Graph는 데이터베이스에 입력한 후 최적 거리 계산 알고리즘에 활용한다.

**4.1.2. 시뮬레이션 결과**



(그림 5) 9번째 하역



(그림 6) 60번째 하역

(그림 5)에서의 경우 4번 신호등에서 최적 거리 계산을 진행할 때 7번 신호등을 경유해서 목적 Node로 가는 경우 해당 경로 내에 다른 차량이 존재하므로 거리 비용이 증가하기 때문에 5번 신호등을 경유해서 가는 경로를 최적 경로로 선정하고 차량이 이동하였다.

반대로 (그림 6)에서의 경우 4번 신호등에서 5번 신호등을 경유해서 가는 경우 해당 경로 내에 다른 차량이 존재하므로 거리 비용이 증가하기 때문에 7번 신호등을 경유해서 가는 경로를 최적 경로로 선정하고 차량이 이동하였다.

실험 결과 차량의 통행량에 따라 경로의 거리 비용(Weight)이 변경되었고, 그에 맞추어 최적 경로를 변경할 수 있었다. 컨테이너를 적재하여 하역할 때까지를 1주기로 정했을 때 1000주기 실험해 본 결과 평균 25% 정도의 대기 시간이 발생하는 것을 확인할 수 있었다.

**4.2. 차량 데이터 수신**

서버에서 계산한 경로 데이터를 데이터베이스에 업데이트 후, 해당 정보를 PHP를 활용하여 불러올 수 있음을 확인하였다. 이러한 정보를 기반으로 자율주행차량에서는 모터 제어문을 작성하여 해당 방향으로 이동함을 확인할 수 있었으며, 신호등에서도 PHP를 통해 데이터를 불러와 LED 모듈에 해당 내용을 시각화할 수 있었다.

**5. 결론**

해당 연구를 통해 스마트 항만 자동화를 이루는 것에 새로운 가능성을 열고, 유동적이며, 저에너지 고효율의 시스템을 개발하는 것을 가능하게 했다. 연구를 통해 센서의 의존도를 낮추고 더 다양한 환경에서 적용될 수 있는 기술을 다양한 국내 항만 상황에 적용할 수 있다는 가능성을 제시하였다.

※ 본 논문은 해양수산부 실무형 해상물류 일자리 지원사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다.

**참고문헌**

[1] 강명규, “네트워크와 알고리즘”, 박영사, 1995  
 [2] J.C.N Climaco, E.Q.V. Martins, “A bicriterion shortest path problem”, European Journal of Operational Research, Vol.11, No.4, pp.399-404, 1982  
 [3] Vedat Akgun, Erhan Erkut, Rajan Batta, “On finding dissimilar paths”, European Journal of Operational Research, Vol.121, No.2, pp.232-246, 2000  
 [4] 김동수, 박만복, “정밀지도의 GPS 데이터를 이용한 최단거리 경로계획 알고리즘”, 한국자동차공학회 춘계학술대회, 2020, p.1012  
 [5] 김용우, 부준필, 김도현, “Dijkstra’s 알고리즘 기반의 능동형 대피 유도 기법”, 한국멀티미디어학회 학술발표논문집, 춘계학술발표논문집, 2008, pp.341-345