

정밀도로지도 경량화 정보 기반 전역경로 생성 가속화 방법론

김 경 현¹⁾ · 홍 석 진¹⁾ · 유 진 우²⁾

국민대학교 자동차공학전문대학원¹⁾ · 국민대학교 자동차IT융합학과²⁾

Accelerated Global-Path Generation Method Based on Lightweight HD-Map Information

Kyeonghyeon Kim¹⁾ · Seokjin Hong¹⁾ · Jinwoo Yoo^{*2)}

¹⁾Graduate School of Automotive Engineering, Kookmin University, Seoul 02707, Korea

²⁾Department of Automotive and IT Convergence, Kookmin University, Seoul 02707, Korea

(Received 19 December 2023 / Revised 26 December 2023 / Accepted 11 January 2024)

Abstract : A global route is an essential technical element in implementing an SAE Level 4 fully autonomous driving system without driver intervention. Under this requirement, research on creating a high-definition, map-based global path and developing a preprocessing algorithm is being actively conducted. In this paper, the BFS (Breadth-First Search) technique was applied by clustering driving route nodes in a precise road map. Later, it was confirmed that the proposed method improved performance after comparing the computational amount and execution time of the Dijkstra algorithm-based global path.

Key words : HD map(정밀도로지도), BFS(너비우선탐색), Lightweight(경량화), Global path generation(전역경로 생성), Automotive driving system(자율주행 시스템)

1. 서론

최근 자동차 산업의 발전에 따라 운전자에게 안전과 편의성을 제공하기 위한 자율주행 기술의 개발 수요가 증가하고 있다. 국내의 자동차 제조사와 연구기관 또는 정부 차원에서의 관련 기술 확보를 위한 투자 및 지원¹⁾에 박차를 가하고 있음을 이에 대한 근거로 들 수 있다. 자율주행 기술은 일반적으로 인지, 판단, 제어의 3가지 단계로 구분할 수 있다. 자율주행 인지란 운전자의 눈을 대신하여 차량 주변 객체 및 도로 환경을 인식하고 이를 자율주행 판단 및 제어에 활용할 수 있는 정보로 전달하는 단계를 의미한다. 이와 관련해 인공지능망 활용²⁾ 및 다중센서의 융합을 통한 인지 성능 개선³⁾에 대한 연구 등이 활발히 이루어지고 있다. 자율주행 판단이란 운전자의 뇌의 역할을 대신하여 인지 정보를 기반으로 자율주행의 관점에서 적절한 상황판단을 수행하는 자율주행 단계를 의미한다. 현재 차량의 주행 상황에서 차선변경, 추월 및 회피 등 적절한 주행전략을 결정하거나 차량이

추종해야 할 전역 및 지역 주행경로를 계획하는 등의 역할을 한다. 이와 관련하여 주변 차량의 안전을 고려한 차선변경을 계획하는 알고리즘 개발⁴⁾, 주행경로 내 장애물과의 충돌을 회피하기 위한 지역경로 계획⁵⁾ 등에 대한 연구가 진행됐다. 자율주행 제어란 판단부에서 도출된 지령에 따라 차량의 가속 및 조향 입력 등의 주행 요소를 조절하여 차량의 실제 주행을 담당하는 기술을 의미한다. 최근 이와 관련하여 강화학습을 적용한 자율주행 제어⁶⁾ 등의 연구가 진행되고 있는 추세이다.

자율주행 연구 분야 중 전역경로 생성은 운전자의 개입을 고려하지 않는 SAE Level 4 완전 자율주행 시스템⁷⁾의 실현을 위해 필수적인 자율주행 판단 기술이다. 이러한 필요성 아래 전역경로를 생성하기 위해 정밀도로지도(HD map; High-Definition map)를 활용한 연구가 활발히 이루어지고 있다. 차량 내 인지 센서는 오염 및 폐색 등으로 인해 오인지/미인지의 인지 성능 저하⁸⁾를 야기할 수 있는 반면, 정밀도로지도는 이에 대한 영향을 전혀 받

*Corresponding author, E-mail: jwyo@kookmin.ac.kr

[†]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.

지 않아 외부 환경 변화에 강건한 자율주행 판단 기능을 구현하기 위해 활용 가능하다는 측면에서 이점을 가진다.

현재 정밀도로지도를 기반으로 한 전역경로 생성 및 전처리 알고리즘 개발 등에 관한 연구가 활발히 이루어지고 있으며 그 예시는 다음과 같다. 홍승우 등⁹⁾은 정밀도로지도 구축과정에서의 제작 오류를 개선하기 위한 전처리 알고리즘 개발 및 이를 활용한 전역경로 생성 방안 관련 연구를 진행하였다. 이원중 등¹⁰⁾은 정밀도로지도 기반 자율주행 테스트베드용 VILS 시스템의 설계 방안을 제안하였고 전역경로 생성을 통해 이를 검증하였다. 하지만, 두 연구 모두 자율주행 차량의 지역 간 이동을 위한 전역경로 생성 시, 정밀도로지도의 범위 확장 시 증가할 연산 비용으로 인한 시간 지연에 대한 해결 방안을 제시하고 있지 않다. 이러한 문제점을 해결하기 위한 목적으로 본 연구에서는 BFS(너비우선탐색)기법을 적용하여 주행경로노드의 군집화를 통한 정밀도로지도 경량화 방법론을 제안하고자 한다. 또한 정밀도로지도 경량화 이후 Dijkstra 알고리즘 기반의 전역경로 생성을 수행하여 제안한 방법론이 기존 방식 대비 연산량 및 실행시간 측면에서 유효한 성능 개선 효과가 있음을 보이고자 한다.

2. 정밀도로지도 기반 전역경로 생성

본 장에서는 정밀도로지도의 유형과 속성, 그리고 이를 활용한 전역경로 생성 알고리즘에 대해 설명하고자 한다.

2.1 정밀도로지도

정밀도로지도는 구축된 데이터의 유형에 따라 점군데이터와 벡터데이터 기반 정밀도로지도로 구분한다. 각 유형의 정밀도로지도는 개발자의 활용 목적 및 응용 분야에 따라 선택적으로 사용할 수 있다.

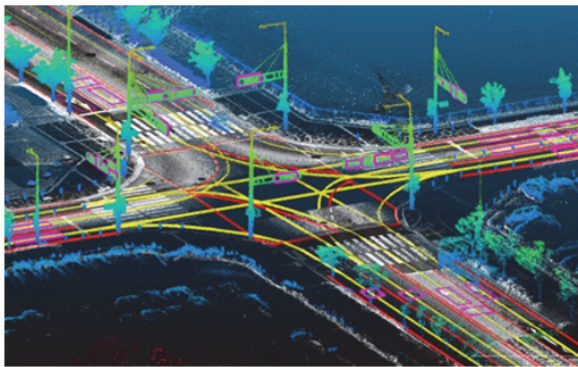


Fig. 1 Point cloud type hd map

2.1.1 점군데이터 기반 정밀도로지도

점군데이터 기반의 정밀도로지도는 Lidar 센서를 기반으로 제작되어 주변 지형지물 및 도로의 표면 등 요소를 3차원 공간 내 점들의 집합으로 표현한다. 점군데이터 기반의 구축 방식은 도로 내 구성 요소의 위치 및 형태 등을 모두 점군의 형태로 보관하여 매우 정확하고 세밀한 정보를 사용자에게 제공한다는 점에서 장점이 있다. 하지만 무수히 많은 양의 데이터를 제공한다는 점에서 데이터 처리 및 저장을 위해 연산량 증가 및 시간 지연 등과 같은 많은 비용을 필요로 한다.

2.1.2 벡터데이터 기반 정밀도로지도

벡터데이터 기반의 정밀도로지도¹¹⁾는 주변 환경 및 지형지물에 관한 정보를 점과 선, 그리고 면과 같은 기하학적 요소로 표현한다. 이러한 특징으로 인해 도로 내 구성 요소들의 좌표 및 속성과 연결 관계 등의 정보를 구조화하여 보관 가능하다. 이러한 특징은 처리 및 저장하기 위한 비용이 비교적 적게 소모된다는 측면에서 이점을 가진다.

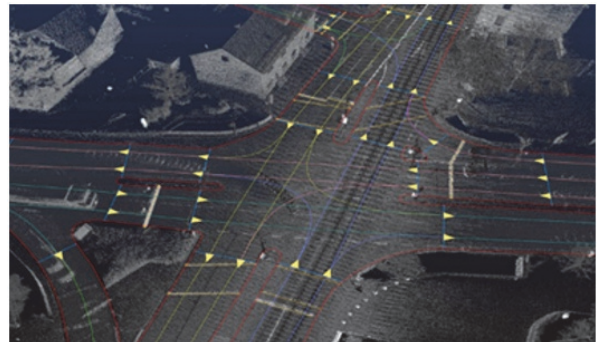


Fig. 2 Vector type hd map

본 연구에서 활용하고자 하는 정밀도로지도는 벡터데이터를 기반으로 제작되어 국토지리정보원¹²⁾에서 구축 및 제공한다. 또한 사용자로 하여금 이를 활용한 개발의 자유도를 부여하기 위해 구축 매뉴얼을 공시하여 벡터데이터 기반 정밀도로지도의 구성 요소 및 속성에 관한 설명을 제공한다.

Table 1은 국토지리정보원의 정밀도로지도 구축 매뉴얼을 참조하여 정밀도로지도의 구성 요소를 표로 작성한 것이다. AI_NODE는 주행경로노드로 도로 및 주행의 운영변환(도로형상 또는 주행환경의 변화)이 발생하는 지점에 생성하는 가상의 점을 의미한다. 주행경로노드는 주행경로링크(A2_LINK)의 시·종점을 의미하며 평면·입체 교차로 시·종점, 교량·터널 시·종점, 요금

Table 1 Components of the hd map of National geographic information institute

1)	A1_NODE
2)	A2_LINK
3)	A3_DRIVEWAYSECTION
4)	A4_SUBDIARYSECTION
5)	A5_PARKINGLOT
6)	B1_SAFETYSIGN
7)	B2_SURFACELINEMARK
8)	B3_SURFACEMARK
9)	C1_TRAFFICLIGHT
10)	C2_KILOPOST
11)	C3_VEHICLEPROTECTIONSAFETY
12)	C4_SPEEDBUMP
13)	C5_HEIGHTBARRIER
14)	C6_POSTPOINT

Table 2 Components of the driving path link used for the purpose of generating a global path

A2_LINK	
	Column name
1)	ID
6)	LinkType
8)	R_LinkID
9)	L_LinkID
10)	FromNodeID
11)	ToNodeID
13)	Length

소 및 회전교차로 내 주행경로링크의 분기·합류 지점으로 유형을 구분한다.

주행경로링크는 진행방향 차로의 중심을 연결한 가상의 선을 의미하며 차량이 주행 가능한 모든 도로 내 연결성을 고려한다. 실제 차량의 유형에 따라, 교차로 내 주행경로, 버스전용차로, 가변차선차로, 일반주행차로 등으로 구분한다.

본 연구에서 전역경로 생성을 목적으로 활용하는 데이터는 A1_NODE와 A2_LINK이다. 두 요소 중, A2_LINK의 속성을 전역경로 생성 알고리즘의 활용 여부에 따라 선정하여 Table 2로 작성하였다.

국토지리정보원의 정밀도로지도 구축 매뉴얼에 의하면 1) ID는 개별 주행경로링크의 구분자를 의미하며, 6) LinkType은 주행경로링크의 유형을 의미한다. 8) R_LinkID와 9) L_LinkID는 대상 주행경로링크의 좌우 차선에 위

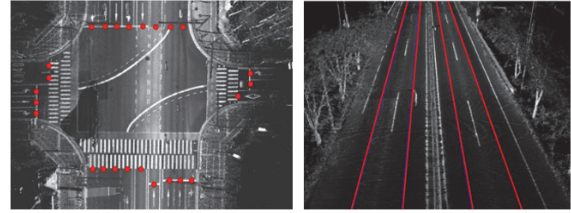


Fig. 3 Driving path node and link of National geographic information institute

치한 주행경로링크의 ID를 의미하며 본 연구에서는 좌우 차선변경 가능한 노드의 연결관계를 정의하기 위해 사용한다. 10) FromNodeID와 11) ToNodeID는 대상 주행경로링크의 시·종점에 위치한 주행경로노드의 ID를 의미하며, 13) Length는 주행경로링크의 길이를 의미한다.

2.2 기존의 전역경로 생성 방법

전역경로 생성이란 차량이 출발지로부터 목적지까지 이동하기 위한 주행경로를 생성하는 기술을 의미한다. 자율주행 차량의 탑승자가 위성항법장치(GPS; Global Positioning System)기반의 목적지 좌표를 입력하면 자율주행 시스템은 현재 위치로부터 목적지까지의 주행경로를 생성하여 이를 추종하는 방식으로 완전 자율주행 시스템을 설계할 수 있다.

전역경로 생성 알고리즘은 대상 주행환경에 따른 입력 데이터의 유형, 활용하고자 하는 용도에 따라 A*,¹³⁾ RRT,¹⁴⁾ Dijkstra¹⁵⁾ 등을 사용한다. A*는 그리드 맵 기반의 전역경로 생성 알고리즘으로 휴리스틱 함수를 정의하여 이를 최소화하는 방향의 최적경로를 탐색한다.

RRT(Rapidly-exploring Random Tree)는 샘플링 기반의 전역경로 생성 알고리즘으로 상태 공간 내 랜덤한 위치에 노드를 생성하며 트리를 성장시키는 과정을 통해 주행환경에 대한 탐색을 수행한다. 다만 해당 알고리즘은 샘플링 기반 알고리즘의 특성 상 실행 시 매번 서로 다른 경로가 생성될 가능성이 있기 때문에 최적의 해를 보장하지 않는다.

Dijkstra는 그래프 내 두 노드 간 최단거리의 경로를 찾기 위해 사용되는 알고리즘으로 본 연구에서 정밀도로 지도를 기반으로 전역경로를 생성하기 위한 목적으로 사용한다. 서로 다른 두 노드를 잇는 링크의 길이를 양의 가중치로 사용하여 주행환경을 그래프로서 정의하기 때문에 실제 이동거리를 고려한다는 점에서 현실 세계에 적용 가능한 알고리즘이다. 이러한 측면에서 정밀도로 지도는 주행경로노드와 주행경로링크로 주행환경을 정의한다는 것을 근거로 해당 연구에서는 Dijkstra 알고리즘을 적용하기에 적합하다고 판단하였다.

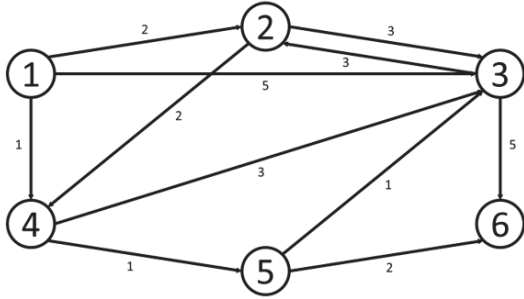


Fig. 4 Weighted graph for defining the connection between all driving path nodes

Dijkstra 알고리즘의 입력은 노드 간 거리를 양의 가중치로 갖는 가중 그래프(Weighted graph)로 정의할 수 있으며 이를 기반으로 주행환경을 묘사할 수 있다. 가중 그래프에 대한 예시를 Fig. 4와 같이 도식화하였다.

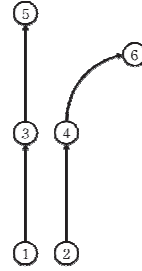
Dijkstra 알고리즘의 동작 단계는 다음과 같다.

- 1) 시작 노드를 제외한 그래프 내 모든 노드까지 거리의 값을 무한대로 초기화한다.
- 2) 노드의 방문 여부를 확인하기 위해 방문한 노드의 집합 배열을 선언한다.
- 3) 그래프 내 모든 노드에 방문할 때까지 하위 과정을 반복한다.
 - A. 저장된 거리의 값이 최소인 노드를 선택한다.
 - B. 선택된 노드에 인접함과 동시에 미방문한 노드를 탐색한다.
 - C. 인접 노드까지의 거리가 현재 계산된 최단거리보다 짧으면 1)의 거릿값을 갱신한다.

Dijkstra 알고리즘은 그래프 내 이동 가능한 경로가 있다면 반드시 최단거리의 경로를 생성하는 최적 알고리즘이다. 하지만 그래프 내 모든 노드에 방문하며 해당 노드에 인접한 모든 노드에 대해 선행 탐색을 수행하기 때문에 알고리즘의 시간복잡도¹⁶⁾는 식 (1)과 같다.

$$\text{Computational Complexity of Dijkstra} = O(N^2) \quad (1)$$

Dijkstra 알고리즘을 활용한 정밀도로지도 기반 전역 경로 생성을 수행하기 위해 주행경로노드와 주행경로링크를 각각 그래프의 노드와 링크로 치환하여 실제 주행환경을 가중 그래프로 묘사하였다. 가중 그래프는 알고리즘 설계의 용이성을 위해 인접 행렬로 정의할 수 있다. Table 2의 정보를 기반으로 주행경로링크의 시·종점을 인접 행렬의 행과 열로 표현하고 이를 잇는 주행경로링



	1	2	3	4	5	6
1	0	3.5	12	0	0	0
2	3.5	0	0	12	0	0
3	0	0	0	3.5	10	0
4	0	0	3.5	0	0	14
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Fig. 5 Converting HD map to adjacency matrix

크의 길이를 행렬의 요소로 구성하였다. 이러한 과정을 통해 주행경로노드 간 연결관계를 인접 행렬의 형태로 정의하였고 그 예시는 Fig. 5와 같다.

노드 1은 노드 2, 노드 3과 연결되어 노드 간 거리를 가중치로 행렬의 요소를 구성한다. 노드 1과 노드 3은 동일한 방향의 주행경로 내 위치하기 때문에 두 노드를 잇는 링크, 즉 노드 1과 노드 3을 시·종점으로 갖는 주행경로 링크의 길이인 12(m)가 행렬의 요소로 삽입된다.

반면, 노드 1과 노드 2는 동일 방향의 주행경로 내 위치하지 않아 주행경로링크를 통한 연결관계에 있지 않으나, 대상 주행경로노드를 시점으로 갖는 주행경로링크의 유형(LinkType)이 6으로 일반차로 내 위치하는 경우, 전역경로 생성 시 차선변경을 고려하는 경로를 생성하기 위해 가상의 주행경로링크를 생성하는 판단 기준으로 사용된다.

만약 해당 주행경로노드를 시점으로 갖는 주행경로링크의 LinkType의 값이 1로 교차로 내 위치하는 링크를 의미한다면 좌·우 주행경로링크의 시점에 해당하는 주행경로노드는 연결관계에 있다고 판단하지 않는다.

그렇지 않은 경우, 즉 LinkType의 값이 6으로 일반차로 내 위치한 주행경로링크의 좌·우 주행경로노드는 차선변경이 가능한 영역으로 두 노드는 연결관계에 있다고 판단한다. 이후 가상의 주행경로링크를 생성 후 이를 인접 행렬의 요소로 삽입한다.

정밀도로지도 내 차선변경에 대한 가중치에 관한 정보는 제공되지 않으나 「도로의 구조·시설 기준에 관한 규칙 제10조」¹⁷⁾의 차로 폭 설계 기준을 참고하여 3.5(m)로 정의한다.

3. 제안하는 정밀도로지도 경량화 방법론

본 장에서는 정밀도로지도 경량화의 필요성 및 그 과정에 관한 설명을 하고자 한다. 앞서 설명한 바와 같이, 정밀도로지도의 주행경로노드 및 주행경로링크를 활용하여 주행환경을 인접행렬로 정의하고 Dijkstra 알고리

즘을 적용하여 전역경로 생성을 수행할 수 있다.

하지만 식 (1)에 의해 Dijkstra 알고리즘의 시간복잡도는 정밀도로지도 내 주행경로노드의 개수가 증가함에 따라 제곱 비례하여 증가하므로 많은 연산 비용을 필요로 한다. 주행경로노드의 개수는 정밀도로지도 대상 지역의 면적 및 차로의 복잡도 등 요소에 의해 결정된다. 차량의 지역 간 이동을 위한 정밀도로지도의 범위 확장 시 제곱 비례하여 증가하는 연산 비용으로 인해 발생하는 시간 지연 등의 현상은 알고리즘의 실시간성 확보에 어려움을 주는 요소이다. 그러므로 본 장에서는 이러한 문제점을 해결하기 위한 정밀도로지도 경량화 방법론을 제안하고자 한다.

3.1 BFS

BFS(Breadth First Search; 너비우선탐색) 알고리즘이란 그래프 혹은 트리와 같은 자료구조의 특정 노드로부터 최인접 노드를 우선으로 방문하며 모든 노드에 대해 선입선출 방식의 Queue 자료구조 기반 탐색을 수행하는 기법이다. 본 연구에서는 주행경로노드에 대한 탐색을 수행하기 위한 목적으로 구형 및 적용 용이성을 근거로 해당 알고리즘을 적용하였지만, 유사한 그래프 탐색 기법으로 깊이우선탐색 또한 적용 가능하다. 노드에 대한 탐색 과정은 다음과 같다.

- 1) 시작 노드를 Queue에 저장한다.
- 2) Queue가 빈 배열이 될 때까지 하위 과정을 반복한다.
 - A. Queue의 최우선 노드를 꺼내어 방문한다.
 - B. 방문한 노드와 인접한 미방문 노드를 모두 Queue에 저장한다.
- 3) 트리 내 모든 노드에 방문하면 탐색을 종료한다.

정밀도로지도를 너비우선탐색 알고리즘에 적용하기 위해 주행경로노드 및 주행경로링크의 속성을 기반으로 구성된 그래프를 인접리스트로 정의하였고 그 예시는 Fig. 6과 같다. Fig. 6의 Table 1열은 주행경로노드의 ID를 의미하고, 2열은 해당 노드에 최인접한 주행경로노드의 ID를 의미한다. Fig. 6의 Table 3열의 Type은 주행경로노드를 잇는 주행경로링크의 유형을 구분한 것으로 1열의 주행경로노드와 2열의 주행경로노드 간 연결관계의 유형으로 정의한다. 연결관계의 유형을 구분한 것은 너비우선탐색 알고리즘 연산 시 전체 노드에 대해 탐색을 수행하며 주행경로노드를 군집화하는 기준으로 사용하기 위함이다. 군집화의 대상은 두 주행경로노드가 교차로 내 위치하지 않고 차선변경이 가능한 경우 Type의 값을

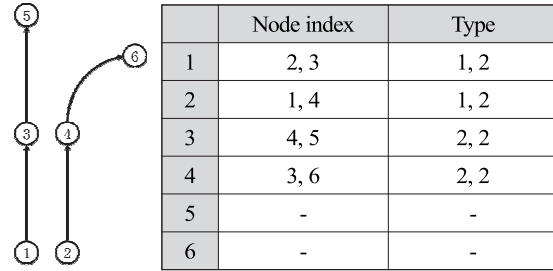


Fig. 6 Converting HD map to adjacency list

1로 정의하여 노드 간 군집을 형성하고 값이 2인 경우 군집화를 수행하지 않는다.

Fig. 6의 Table 1행을 예시로 노드 1과 노드 3은 동일한 주행방향 차선 내 있어 군집화 대상이 아니므로 노드 간 Type을 2로 정의한다. 노드 1과 노드 2는 두 노드를 시점으로 갖는 주행경로링크의 LinkType이 6으로 일반차로 유형이며 좌우로 차선변경이 가능한 구간이기 때문에 노드 간 Type을 1로 하여 군집화 대상으로 구분한다.

이후, 정의된 인접리스트를 BFS 알고리즘에 적용하여 정밀도로지도 내 모든 노드에 대해 탐색하며 연결 유형이 1인 노드에 대해 군집화를 수행한다.

3.2 제안하는 경량화 알고리즘의 구조

본 절에서는 정밀도로지도 경량화 기반 전역경로 생성 알고리즘의 구조를 설명하고자 한다. 기존 전역경로 생성 방식은 주행경로노드 및 주행경로링크를 이용해 Weight matrix를 정의하고 이를 입력으로 Dijkstra 알고리즘에 적용하여 출발지로부터 목적지까지 이동하기 위한 최단경로를 생성한다.

본 연구에서 제안하는 정밀도로지도 경량화 기반 전역경로 생성 알고리즘의 구조는 Fig. 7과 같이 도식화할 수 있고 그 과정은 다음과 같다.

- 1) 주행경로노드를 너비우선탐색 기반 군집화하여 Cluster 단위로 경량화한 정밀도로지도를 정의한다.
- 2) 정밀도로지도 경량화 후, Dijkstra 알고리즘을 적용하여 Cluster 단위로 구성된 전역경로를 생성한다.
- 3) 생성된 Cluster 단위의 경로를 구성하는 주행경로노드에 대해 Dijkstra 알고리즘을 다시 적용하여 Node 단위의 전역경로를 생성한다.

기존 방식 대비 제안한 경량화 방법론을 통한 전역경로 생성 방안의 성능 개선 효과를 비교하기 위해, 다음과 같이 예시를 통해 설명하고자 한다.

만약 주행경로노드의 개수가 1,000개인 정밀도로지도

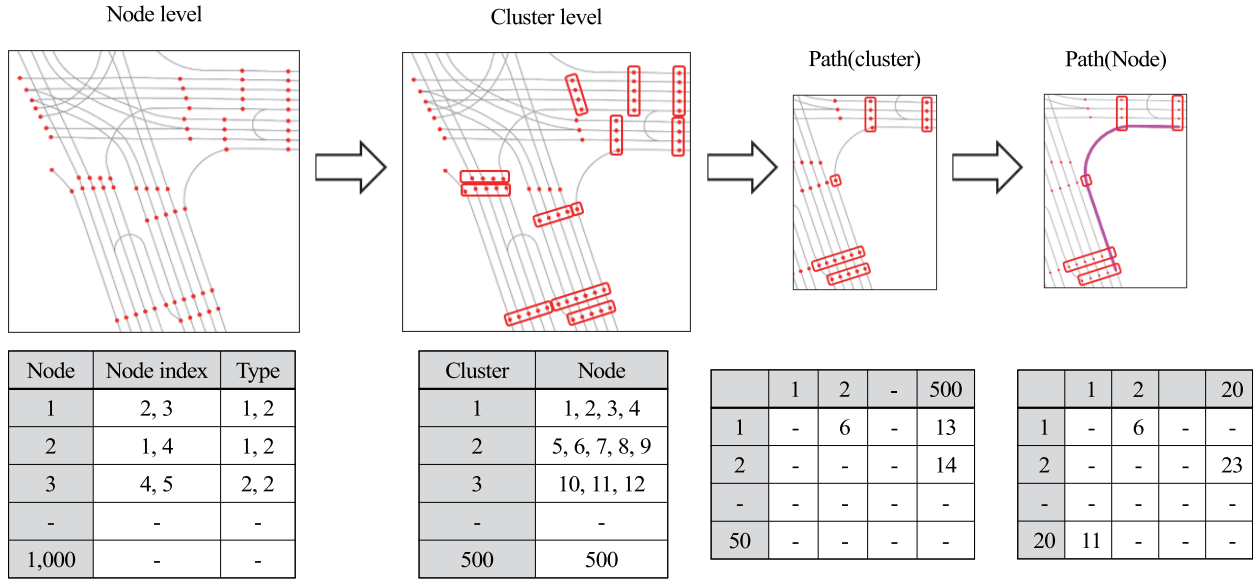


Fig. 7 Proposed architecture

Table 3 Comparing the between conventional and proposed methodology

	Original	Proposed
1	1,000,000	250,000
2	0	10,000
Sum	1,000,000	260,000
Reduction rate	74 %	

에 기존의 전역경로 생성 방식을 적용한다면, 식 (1)을 적용하여 단순 계산한 연산 비용은 Table 3의 2열과 같이 추정할 수 있다.

하지만 본 연구에서 제안하는 방안은 다음과 같다. 경량화를 통해 획득한 500개의 군집으로 구성된 정밀도로 지도를 재정의하고 이에 전역경로 생성을 수행하여 군집 단위 경로를 도출한다. 이후, 군집 단위 경로 내 포함하는 주행경로노드 100개에 대해 전역경로를 재생성한다면 차량이 실제 추종해야 할 주행경로노드 단위의 주행 경로를 생성할 수 있다. 이에 대한 연산량 단순 계산 방식은 Table 3의 3열과 같고, 알고리즘의 연산량 감소율은 74%이다.

실험하고자 하는 개발 환경에 따라 Runtime의 수치는 상이할 수 있으나, 연산량 감소율과 유사한 경향으로 단축됨을 확인할 수 있다.

4. 실험 결과

제안한 알고리즘의 성능 개선 효과를 검증하기 위해 3

개 지역의 정밀도로 지도를 대상으로 실험을 진행하였다. 개선 효과를 검증하기 위한 과정은 다음과 같다.

- 1) 대상 지역 정밀도로 지도의 주행경로노드 감소율을 계산하여 정밀도로 지도의 경량화 정도를 수치적으로 확인한다.
- 2) 기존 및 제안한 방식 모두 동일한 출발·목적지점을 갖는 동일 경로를 생성하는 데 드는 연산시간 및 시간 복잡도를 계산하여 알고리즘의 연산 비용에 대한 비교를 수행한다.

4.1 정밀도로 지도 경량화 결과 비교

Fig. 8은 경기도 화성시 소재의 자동차안전연구원 주행시험장(K-city)에 경량화를 수행한 결과로 기존 주행

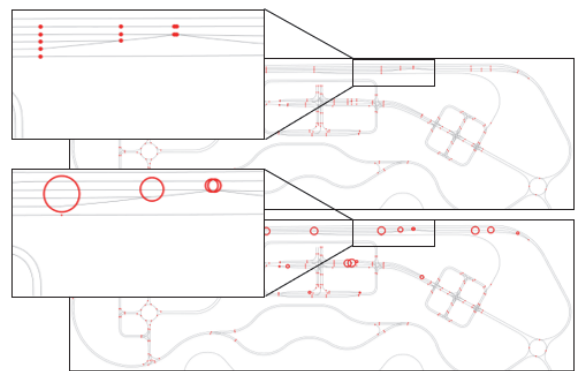


Fig. 8 Lightweighted K-city

Table 4 Comparing the # of nodes (K-city)

	# of Nodes
Original	712
Proposed	622
Reduction rate	12.64 %

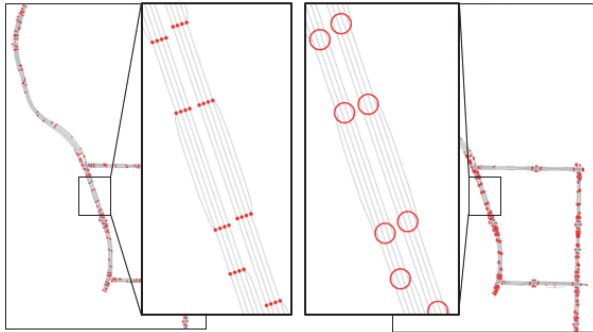


Fig. 9 Lightweighted Zero-city

Table 5 Comparing the # of nodes (Zero-city)

	# of Nodes
Original	1,102
Proposed	561
Reduction rate	49.09 %

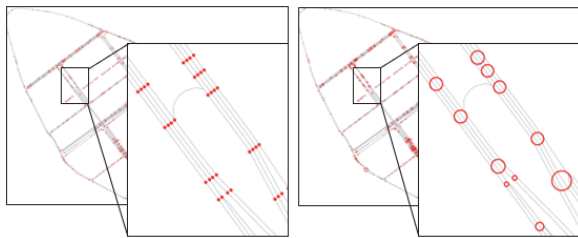


Fig. 10 Lightweighted Yeouido

Table 6 Comparing the # of nodes (Yeouido)

	# of Nodes
Original	2,042
Proposed	1,027
Reduction rate	49.71 %

경로노드 개수 712 개에서 제안 알고리즘 적용 후 622 개로 12.64 % 감소하는 것을 확인하였다.

Fig. 9는 경기도 판교시 소재의 자율주행 테스트베드 Zero-city에 경량화를 수행한 결과로 기존 주행경로노드 개수 1,102개에서 제안 알고리즘 적용 후 561개로 49.09 % 감소하는 것을 확인하였다.

Fig. 10은 서울시 여의도 자율주행 테스트베드에 경량화를 수행한 결과로 기존 주행경로노드 개수 2,042 개에서 제안 알고리즘 적용 후 1,027 개로 49.71 % 감소하는 것을 확인하였다.

4.2 전역경로 생성 결과 비교

해당 실험은 CPU: AMD Ryzen 7 3800XT / RTX 3070 / RAM : 32GB의 환경에서 진행되었다. Matlab 2019b를 이용하여 제안한 알고리즘을 검증하였고, 각 알고리즘의 Runtime을 측정하였다.

기존 방식과 제안한 방법론 모두 동일한 Dijkstra 알고리즘을 적용하여 알고리즘의 시간복잡도의 차이는 변화가 없다. 그러므로 각 알고리즘의 연산량의 계산은 식 1)의 시간복잡도를 적용하여 도출할 수 있다. 다만 Table 3에서 기술한 바와 같이 기존의 전역경로 생성 방식의 연산량은 전체 노드의 개수의 제곱한 값을 가지나, 제안한 방식에서는 1차 전역경로 생성에서 도출된 Cluster의 개수의 제곱과 2차 전역경로 생성에서 사용되는 Cluster 내 주행경로노드의 개수의 제곱한 값을 합산한다는 점에서 차이가 있다.

3개 지역 정밀도로지도에 각각 6개의 동일한 경로를 생성하는 시나리오에 대해 제안하는 알고리즘의 적용 전과 후 결과를 비교하였다. 각 지역 별 6개 시나리오의 선정 근거는 알고리즘의 유효성을 검증하기 위해 차선 변경, 좌·우 선회 및 유턴경로 등을 포함하는 복잡한 주행경로를 기준으로 결정하였다.

Table 7은 자동차안전연구원 주행시험장 K-city 대상 실험 결과로 평균 Runtime 감소율은 7.22 %, 평균 연산량 감소율은 23.50 %임을 확인하였다.

Table 8은 판교 Zero-city를 대상으로 실험을 진행한 결과로 평균 Runtime 감소율은 46.08 %, 평균 연산량 감소율은 73.08 %임을 확인하였다.

Table 9는 여의도 자율주행 테스트베드를 대상으로 실험한 결과로 평균 Runtime 감소율은 64.74 %, 평균 연산량 감소율은 74.51 %임을 확인하였다.

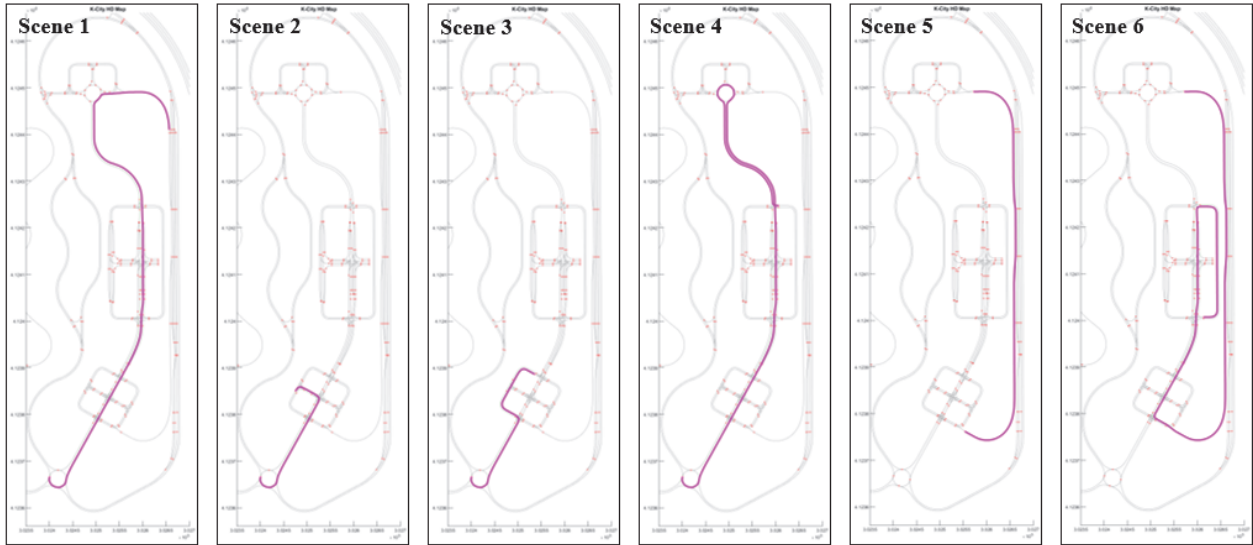


Fig. 11 K-city global path generation scenario

Table 7 Results of K-city scenario

Scene		Runtime(sec)	Computational complexity
1	Original	0.01467	506,944
	Proposed	0.01305	388,328
	Reduction rate	11.09 %	23.40 %
2	Original	0.01488	506,944
	Proposed	0.01270	386,984
	Reduction rate	14.63 %	23.66 %
3	Original	0.01434	506,944
	Proposed	0.01320	386,948
	Reduction rate	7.91 %	23.67 %
4	Original	0.01499	506,944
	Proposed	0.01396	388,484
	Reduction rate	6.90 %	23.37 %
5	Original	0.01329	506,944
	Proposed	0.01270	387,245
	Reduction rate	4.39 %	23.61 %
6	Original	0.01302	506,944
	Proposed	0.01323	389,000
	Reduction rate	-1.61 %	23.27 %
Mean		7.22 %	23.50 %



Fig. 12 Zero-city global path generation scenario

Table 8 Results of Zero-city scenario

Scene		Runtime(sec)	Computational complexity
1	Original	0.03260	1,214,404
	Proposed	0.01441	321,121
	Reduction rate	55.79 %	73.56 %
2	Original	0.03062	1,214,404
	Proposed	0.01785	328,410
	Reduction rate	41.69 %	72.96 %
3	Original	0.03088	1,214,404
	Proposed	0.01888	336,330
	Reduction rate	38.85 %	72.30 %
4	Original	0.03213	1,214,404
	Proposed	0.01743	324,922
	Reduction rate	45.77 %	73.24 %
5	Original	0.03167	1,214,404
	Proposed	0.01463	318,690
	Reduction rate	53.80 %	73.76 %
6	Original	0.03097	1,214,404
	Proposed	0.01841	332,145
	Reduction rate	40.57 %	72.65 %
Mean		46.08 %	73.08 %

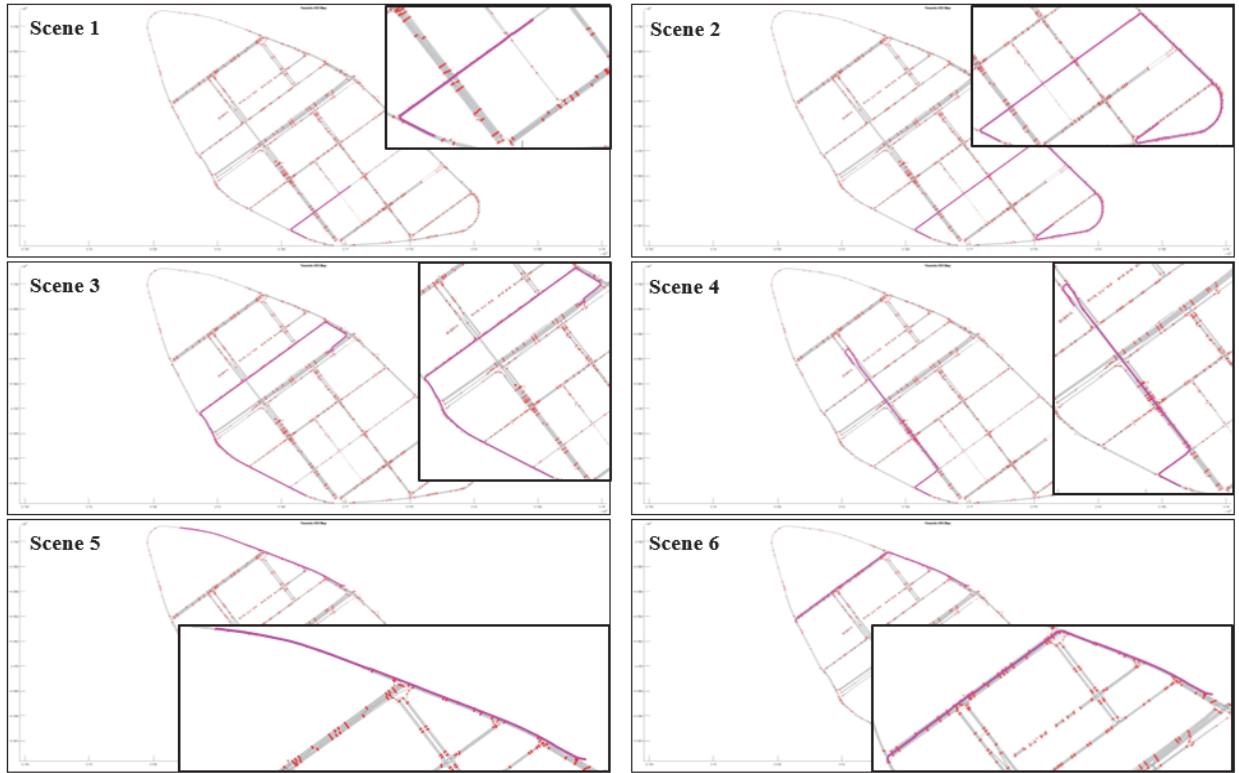


Fig. 13 Yeuido global path generation scenario

Table 9 Results of Yeuido scenario

Scene		Runtime(sec)	Computational complexity
1	Original	0.16822	4,169,764
	Proposed	0.04311	1,055,513
	Reduction rate	74.37 %	74.69 %
2	Original	0.16297	4,169,764
	Proposed	0.05012	1,066,393
	Reduction rate	69.24 %	74.43 %
3	Original	0.14751	4,169,764
	Proposed	0.05412	1,065,545
	Reduction rate	63.31 %	74.45 %
4	Original	0.14815	4,169,764
	Proposed	0.04914	1,067,050
	Reduction rate	66.83 %	74.41 %
5	Original	0.14397	4,169,764
	Proposed	0.05798	1,056,250
	Reduction rate	59.73 %	74.67 %
6	Original	0.14249	4,169,764
	Proposed	0.06421	1,067,498
	Reduction rate	54.93 %	74.40 %
Mean		64.74 %	74.51 %

5. 결론

본 연구에서는 전역경로 생성의 가속화를 위한 정밀도로지도 경량화 알고리즘을 설계하였다. 주행경로노드를 속성에 따라 군집화하여 정밀도로지도를 경량화하였고 전역경로 생성을 수행하여 기존의 방식 대비 제안한 방안이 성능 개선의 효과가 있음을 확인하였다. 3개 지역의 정밀도로지도에 동일한 경로를 생성하는 각 6개 시나리오를 비교한 결과, 연산 비용 및 시간복잡도 감소를 확인하여 본 연구에서 제안하는 경량화 방안의 유효성을 검증하였다. 현재 국토지리정보원에서 제공하는 정밀도로지도는 지역별로 구축되어 개별 사용할 수 있도록 제공되고 있다. 하지만 향후 지역 간 이동 목적의 전역경로를 생성하기 위해 기존의 정밀도로지도를 통합¹⁸⁾하여 사용하는 경우 많은 양의 주행경로노드를 알고리즘의 입력으로 사용하여야 한다는 점에서 실시간성 확보 여부는 매우 중요한 요소이다.

이러한 점에서 본 연구에서 제안하는 경량화 알고리즘의 적용은 확장된 범위의 정밀도로지도를 기반으로 전역경로 생성을 수행하기 위해 연산량 및 실행시간 단축을 통해 실시간성을 확보하기 위한 수단을 제공한다는 측면에서 효과를 기대하는 바이다.

후 기

이 논문은 2024년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임(P0020536, 2024년 산업혁신인재성장지원사업).

References

- 1) K. Hong, and W. Son, "Research Trends of Vehicle-In-the-Loop-Simulation for Automated Driving," Auto Journal, KSAE, pp.21-25, 2022.
- 2) N. Sharma, R. Sharma and N. Jindal, "Machine Learning and Deep Learning Applications-A Vision," Global Transitions Proceeding 2, pp.24-28, 2021.
- 3) F. Garcia, D. Martin, A. Escalera, and J. Armingol, "Sensor Fusion Methodology for Vehicle Detection," IEEE Intelligent Transportation Systems Magazine, Vol.9, pp.123-133, 2017.
- 4) T. Oh, W. Son, T. Ahn, Y. Lee, and K. Park, "Development of Automated Lane Change Algorithm Considering Safety of Surrounding Vehicles," Transactions of KSAE, Vol.29, No.5, pp.391-405, 2021.
- 5) Y. Kim, Y. Shim, K. Min, S. Lee, and H. Son, "Decision Making and Path Planning for an Autonomous Vehicle Conducting Lane Changing, Obstacle Avoidance, and Waypoint Following," IEIE Annual Conference Proceeding, pp.510-514, 2020.
- 6) A. Yu, R. Smith and R. Bedi, "Deep Reinforcement Learning for Simulated Autonomous Vehicle Control," Stanford University, StuDocu, 2016.
- 7) SAE J3016, Levels of Driving Automations, <https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic>, 2019.
- 8) J. Seong, Y. Yoon, and D. Han, "Deep Learning-Based Camera Sensor Failure Prediction Model," KICS Winter Conference Proceeding, pp.338-339, 2022.
- 9) H. Hong, W. Son, K. Park, S. Kwun, I. Choi, and S. Cho, "Design of Preprocessing Algorithm for HD-Map-Based Global Path Generation," The Korea Institute of Intelligent Transport Systems, Vol.21, No.1, pp.273-286, 2022.
- 10) W. Lee, and S. Kee, "Implementation of VILS Systems for Autonomous Driving Testbed Based on HD Map," Transactions of KSAE, Vol.31, No.7, pp.503-511, 2023.
- 11) Y. Na, S. Kim, Y. Kim, J. Park, J. Jeong, K. Jo, S. Lee, S. Cho, M. Sunwoo, and J. M. Oh, "HD Map Usability Verification for Autonomous Car," Transactions of KSAE, Vol.28, No.11, pp.797-808, 2020.
- 12) NGII, HDmap Constrection Manual, https://www.ngii.go.kr/kor/contents/view.do?sq=1195&board_code=contents_data, 2019.
- 13) F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, and L. Jurisica, "Path Planning with Modified a Star Algorithm for a Mobile Robot," Procedia Engineering, Vol.96, pp.59-69, 2014.
- 14) I. Noreen, A. Khan, and Z. Habib, "A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms," International Journal of Computer Science and Network Security, Vol.16, No.10, 2016.
- 15) H. Wang, and Y. Yu, "Application of Dijkstra Algorithm in Robot Path-Planning," International Conference on Mechanic Automation and Control Engineering, pp.1067-1069, 2011.
- 16) Wikipedia, https://en.wikipedia.org/wiki/Time_complexity, 2021.
- 17) Korea Ministry of Government Legislation, Rules on Road Structure and Facility Standards, 2021.
- 18) M. Kim, Chosun Biz, https://biz.chosun.com/policy/policy_sub/2022/09/20/FS5JGTSPJBBOTAE7MSTTTOMTQU/, 2022.