

# 그래프 구조를 이용한 도로 네트워크 갱신 방안

## A Study on Update of Road Network Using Graph Data Structure

강 우 빈\* · 박 수 홍\*\* · 이 원 기\*\*\*

\* 주저자 : 인하대학교 공간정보공학과 석사과정

\*\* 교신저자 : 인하대학교 공간정보공학과 교수

\*\*\* 공저자 : 인하대학교 공간정보공학과 학사과정

Woo-bin Kang\* · Soo-hong Park\*\* · Won-gi Lee\*\*\*

\* Dept. of Geoinformatic Eng., Univ. of Inha

\*\* Dept. of Geoinformatic Eng., Univ. of Inha

\*\*\* Dept. of Geoinformatic Eng., Univ. of Inha

† Corresponding author : Soo-hong Park, spark@git.inha.ac.kr

Vol.20 No.1(2021)

February, 2021

pp.193~202

pISSN 1738-0774

eISSN 2384-1729

[https://doi.org/10.12815/kits.](https://doi.org/10.12815/kits.2021.20.1.193)

2021.20.1.193

Received 8 October 2020

Revised 22 October 2020

Accepted 17 February 2021

© 2021. The Korea Institute of Intelligent Transport Systems. All rights reserved.

### 요 약

고정밀 지도의 갱신은 정사영상 또는 점군 데이터 등을 원천 자료로 하여 기하 정보를 우선적으로 수정한 이후 지도를 구성하는 공간객체들 간의 연관관계를 재정립하는 방식으로 진행된다. 이러한 일련의 과정들은 기하 정보를 처리하는 데에 많은 시간을 소요하므로 차량의 실시간 경로 계획(Real-time route planning)에 빠르게 적용되기 어렵다. 따라서 이 연구에서는 그래프 구조를 활용하여 경로 계획을 위한 도로 연결구조를 우선적으로 업데이트 하는 방식 및 도로 네트워크의 특징을 고려한 그래프 구조의 저장 유형을 제안하였다. 또한 제안된 방법을 실제 도로 자료에 적용해 봄으로써 실시간 경로 정보 전송 시의 활용 가능성에 대해 검토하였다.

핵심어 : 지도 업데이트, 그래프 구조, 실시간 경로 정보, 도로 네트워크

### ABSTRACT

The update of a high-precision map was carried out by modifying the geometric information using ortho-images or point-cloud data as the source data and then reconstructing the relationship between the spatial objects. These series of processes take considerable time to process the geometric information, making it difficult to apply real-time route planning to a vehicle quickly. Therefore, this study proposed a method to update the road network for route planning using a graph data structure and storage type of graph data structure considering the characteristics of the road network. The proposed method was also reviewed to assess the feasibility of real-time route information transmission by applying it to actual road data.

Key words : Map Update, Graph Data Structure, Real-time Route Information, Road Network

## I. 서 론

### 1. 개요

자율주행 차량의 센서 인지범위는 일반적으로 최대 200m 정도로 알려져 있지만 도로에서 흔히 발견할 수 있는 표지판, 신호등과 같은 작은 시설물들은 수십 m 정도에서 인지함으로써 급제동이 불가피하고, 작은 곡선부, 시야가 차단된 교차로 등의 경우 인지범위가 제한적이기 때문에 센서 시스템에 의존하는 형태로는 원활한 자율주행을 구현할 수 없다. 따라서 현재 세계 각국에서 이를 보완하기 위해 고정밀 지도의 제작과 관련한 연구가 계속해서 수행되고 있으며, 고정밀 지도의 경로 정보를 실시간으로 전송하기 위한 업계 표준인 ADASIS<sup>1)</sup> 등이 사용되고 있다.

고정밀 지도의 정보는 드론이나 차량을 이용한 MMS<sup>2)</sup>나 개인 사용자의 휴대 기기 등 다양한 플랫폼을 이용하여 수집 및 갱신이 가능하지만 변경된 지도 정보를 갱신하기 위해서는 수집된 지도의 기하 정보를 반영한 뒤, 변경된 공간 객체들 간의 상관관계인 위상 구조(Topology)를 나중에 재정립하는 구조가 일반적이다.

현재 지도 갱신 관련 연구동향을 살펴보면, 지도를 구성하는 기하 정보, 위상 구조 및 속성정보 모두를 일시에 업데이트 하는 것에 중점이 맞춰져 있다. 따라서 사용되는 원천 자료 또한 지상의 여러 객체에 대한 통합적인 정보를 수집할 수 있는 원격탐사 자료, 항공영상 자료, LiDAR<sup>3)</sup> 점군(Point Cloud) 자료들이 대부분이다. 관련 연구 동향 또한 위의 자료들에서 객체 추출 및 기하 정보를 보정하여 기존의 지도 데이터에 객체 단위 매칭을 이용하는 형태로 지도를 갱신하고 있다(Seol et al., 2019).

이와 관련한 선행 연구로는 Beyen et al.(2012)이 국가 지도 기관의 데이터베이스 갱신 작업의 자동화 관련 분야의 연구가 추가적으로 필요하다고 언급하면서, 아웃소싱 작업 결과물의 시각적 검수 자동화에 관한 연구를 수행하였다. Jomrich et al.(2017)은 고정밀지도의 지속적인 업데이트 스트림을 효율적으로 제공하기 위한 상황별 지도 데이터 전송 통신 규약에 관하여 연구하였다. Kim et al.(2017)은 무인항공사진측량을 이용하여 대상지의 정사영상을 생성하고, 이를 기반으로 수치지도의 2D 레이어 추출, 점군 데이터를 통한 DEM 등으로 수치지도 수정 및 갱신을 수행하는 방안을 도출하였다. Ga et al.(2008)은 차량용 내비게이션 시스템의 최신성이 떨어지는 점을 네트워크 데이터의 노드, 링크마다 버전 정보를 부여하여 전송되는 경로 검증을 수행해 해결하는 방안을 제시하였다.

선행연구들에서 공통적으로 사용했던 측량 이후 측량 성과물을 검증하고 검증된 성과물을 지도에 반영하여 타일 단위로 갱신하는 프로세스는, 정확하지만 필연적으로 처리시간이 오래 소요된다는 문제점이 있다. 지도 정보의 갱신 자동화 시스템이 연구되고 있으나 이 방식 또한 결국 작업자의 육안을 통해 실제 도로 형태와 비교하여 오차, 또는 구조적 오류 등을 검수하는 과정을 거치지 않으면 정확도가 크게 떨어지게 된다. 이러한 문제점 때문에 실질적으로 갱신 정보가 수집되고 지도에 적용되기까지는 많은 시간이 소요된다. 또한, 서버(Server)-클라이언트(Client) 간의 구조에서 이미 작성된 형태의 지도를 전송받는 경우 타일(Tile) 단위의 정보를 통합적으로 수신하며, 차량의 고속 주행 시 무선 통신 상황에 크게 영향을 받는다. 이러한 지도 갱신의 한계 때문에 차량의 고속 이동과 같은 상황에 차량이 가지고 있는 지도 정보와 서버 등의 중앙 처리 센터에서 가지고 있는 지도 정보가 상이한 경우, 차량 또는 서버의 지도 정보의 빠른 갱신이 어려울 수밖에

1) ADASIS : Advancing map-enhanced Driver Assistance Systems Interface Specifications, 첨단 운전자 보조 시스템

2) MMS : Mobile Mapping System, 모바일 맵핑 시스템

3) LiDAR : Light Detection And Ranging, 레이저 기반 사물 감지 및 거리 측정 기술

없다. 자율주행 차량과 같은 경우, 안전성을 확보하기 위해 차량 센서 보완의 목적으로 고정밀 지도를 활용하기 때문에 이러한 문제가 더욱 두드러진다. 기존의 지도 갱신 방식은 측량 성과를 기반으로 해당 정보를 적용 및 검수하는 작업을 거치기 때문에 정확하고 도로망과 같은 차량 주행과 직접적으로 관련된 부분 외에도 POI<sup>4)</sup> 또는 기타 지형, 지물에 대한 정보를 함께 포함할 수 있다. 그러한 장점에도 불구하고 과정 자체에 소요되는 시간 문제로 인해 갱신의 주기가 최소 1개월에서 길게는 수 년 단위로 이루어진다.

이와 같은 한계점을 보완하기 위해, 그래프 구조를 적용하여 지도를 나타내고 지도의 그래프 구조만을 우선적으로 갱신하는 방식을 실험해보고자 한다. 이 방식을 사용하면 지도를 구성하는 교차로, 도로 등의 연결관계를 먼저 갱신하게 되며 이 경우 측량 성과물을 통해 기하 정보를 우선적으로 갱신하고 이후 위상구조를 기하 구조에 맞게 수정하는 방식보다 갱신 정보가 반영되기까지의 시간을 상대적으로 적게 소모한다.

따라서 해당 방식은 도로망과 같이 구조 자체가 더욱 중요한 지도 데이터에 적용이 간편한 장점이 있다. 물론 지도의 기하 정보가 상대적으로 느리게 갱신되지만 차량의 실시간 이동 경로 정보 생성과 같은 서비스 등에 사용하는 경우 타일 단위의 지도 업데이트를 수행하는 기존 방식보다 간단하게 적용할 수 있다.

## II. 연구 방법

이 연구에서는 상술한 대로 기존 측량성과 기반의 지도 업데이트 방식을 대신하여, 그래프 구조를 활용한 지도 업데이트 방식을 간략히 실험해보고자 한다. 먼저, 단순화된 형태의 도로망 데이터 갱신 시나리오를 작성하고, 작성된 시나리오를 바탕으로 도로망 데이터에 적합한 형태로 그래프 구조를 작성하여 실제 도로망 데이터를 활용한 위상 구조 갱신이 실효성이 있는지 살펴보고자 한다.

### 1. 도로망 그래프 구조

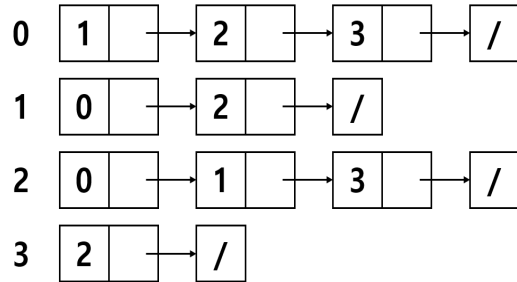
흔히, 도로망 구조를 표현하기 위해 그래프 구조를 이용한다. 그래프 구조는, 노드(Node)와 노드 사이를 잇는 간선(Edge)의 집합을 이용해 연결되어 있는 객체 간의 관계를 나타내는 구조이다. 그래프 구조는 이러한 특징 때문에 도로망을 직관적으로 표현하기 쉽다. 도로망에서의 교차로는 그래프 구조에서의 노드가 되며, 교차로와 교차로를 잇는 도로는 간선으로 대체할 수 있기 때문이다. 따라서 Dijkstra 알고리즘이나 A-star 알고리즘과 같은 경로 탐색을 위한 방법론들을 적용하기 위해 경로 이동시의 비용(Cost)을 구현하기 적합한 가장 방향 그래프 등을 도로망 표현에 사용하는 것이 실제계 표현에 유리한 장점이 있다.

이러한 장점 때문에 도로망 형태를 표현하는 데에 그래프 구조를 사용하려면 우선적으로 그래프 구조의 저장형태에 따른 이해가 필요하다. 그래프 구조는 크게 인접 행렬과 인접 리스트의 두 가지 형태로 저장할 수 있다. 인접 행렬은,  $V$ 개의 노드를 갖는 그래프를 0과 1의 값을 갖는  $V \times V$  행렬로 표현하는 방법이다. 셀이 1의 값을 가지는 경우 노드가 연결된 상태임을 나타낸다. 아래 <Fig. 1, 2>는 인접 행렬과 인접 리스트의 간단한 모식도이다.

4) POI : Point Of Interest, 관심 지점

	0	1	2	...	V-1	V
0	0	1	0		0	1
1	1	0	0		1	0
2	0	0	0		1	1
⋮					0	0
V-1	1	1	0	1	0	0
V	1	0	0	1	1	0

<Fig. 1> Adjacency matrix example



<Fig. 2> Adjacency list example

인접 행렬은 행렬을 이용함으로써 노드 간의 연결관계를 빠른 시간 내에 확인할 수 있다는 것이 장점이나, 실제 연결된 정점 수와 관계없이 연결된 정점을 찾는 데 시간이 많이 소요되는 점과 연결되지 않은(간선이 없는) 형태를 표현하는 데에도 공간을 할당하는 점이 비효율적이다.

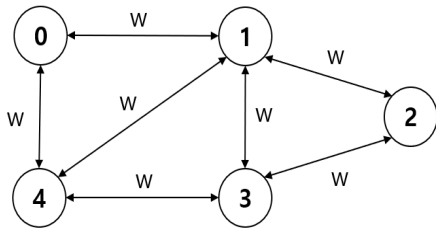
인접 행렬과는 다른 방법으로, 각각의 정점에 대하여 인접한 정점 번호를 저장하는 인접 리스트가 있다. 인접 리스트는 연결 리스트(Linked List)를 이용해서 구현하며 주로 벡터(Vector)와 같이 길이를 변경할 수 있는 배열을 이용하여 구현한다. 인접 리스트는 위의 예시와 같이 노드(좌측 0, 1, 2, 3)에 연결된 다른 노드들이 연결 리스트 형태로 저장된다. 고정된 크기의 행렬을 사용하는 인접 행렬에 비해 초기 작성 시 구현 시간이 많이 소요되나, 특정 노드에서 연결된 다른 노드를 모두 찾는 데 필요한 만큼만 검색하게 되며 동적 배열을 이용하기 때문에 필요한 만큼만 공간을 활용하는 것이 장점이다. 그러나 인접 행렬이 노드간의 연결 확인이 셀 안의 값을 읽는 형태로 비교적 간단한 반면에 인접 리스트는 노드간의 연결을 확인하려면 연결된 노드를 전부 확인해야 한다는 단점이 있다.

인접 행렬과 인접 리스트는 각각 장단점이 존재하지만 도로망의 특성상 한 개의 노드에 많은 간선이 존재하지 않는 희소 그래프이므로 인접 리스트를 사용하는 것이 바람직하다고 볼 수 있다. 또한, 인접 행렬은 도로망을 구성하는 노드들의 변경이나 생성 / 삭제가 어려운 형태로 작성되므로 연구의 목적인 도로망 갱신을 위한 구조에는 인접 리스트가 상대적으로 적합하다.

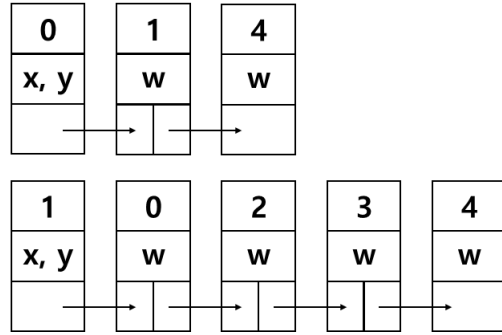
## 2. 인접 리스트 적용

앞서 작성된 바와 같이 인접 리스트를 활용하는 형태로 그래프를 구성하였다면, 이후 작성된 노드들에 대한 수정 및 삭제를 통해 검색, 변경, 삽입, 삭제 등과 같은 지도 자료의 유지·보수 작업을 수행하게 된다. 연결 리스트를 구성하는 방식은 단일 연결 리스트, 이중 연결 리스트, 다중 연결 리스트, 원형 연결 리스트의 네 가지 방식이 대표적이다.

노드 다섯 개를 가진 도로망의 구조를 가중치(w)를 가진 양방향 그래프 형태로 도식화하면 다음 <Fig. 3>과 같이 작성할 수 있다. 아래 <Fig. 4>는 <Fig. 3>에 제시된 그래프의 인접 리스트 중 노드 0, 1에 해당하는 부분만을 간략하게 나타낸 것이다.



<Fig. 3> Road network example

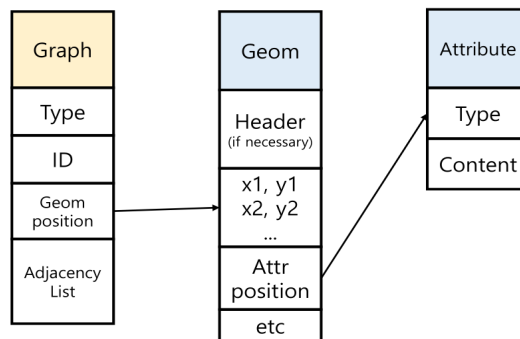


<Fig. 4> Linked list

<Fig. 4> 에서의 연결 리스트 방식은, 각 노드가 첫 번째 노드로부터 단일 방향의 참조를 하는 형태인 단일 연결 리스트 방식이다. 각 노드가 가진 기하 정보는 x,y 좌표값의 형태로 헤드에 저장되며 다른 노드로 연결할 시 주어지는 가중치와 같은 정보들이 리스트에 저장되는 형태로 구성된다. 도로망의 형태나 목적에 따라 앞서 언급한 다양한 연결 리스트 방식을 이용하여 인접 리스트를 구성할 수 있으며, 이러한 차이점으로부터 접근, 변경, 삭제 와 같은 질의 처리의 방식이 달라진다. 이 연구에서는 도로망의 연결관계를 표현하기에 적합하고 가장 간단한 형태이자 사용하려는 예시 자료 및 목적에 부합하는 단일 연결 리스트 구조를 사용하였다.

### 3. 파일 시스템 적용

실제 차량 임베디드 시스템에 그래프 구조 기반의 지도를 탑재한다는 가정을 할 때, 지도 정보의 분할 저장 이 필수적이다. 왜냐하면, 차량에 탑재되는 단말의 성능은 일반적인 컴퓨팅 환경보다 낮을 수 밖에 없으며, 따라서 저장되는 지도 정보 또한 저장 용량에 처리하기 쉬운 형태여야 한다. 인접 리스트 구조에 지도 정보를 포함 하려면 구조 자체에 기하 정보를 포함하여야 해서 뼈대가 되는 위상 구조의 형태를 나타내는 데에 많은 용량을 차지하며, 노드의 추가/삭제/변경 시 해당 정보를 동시에 업데이트 해주어야만 하는 문제점이 있다. 따라서, 기하 정보 및 지도에 삽입되어야 하는 정보를 별도의 파일 형태로 저장하여 기하 정보의 의존성을 낮추고, 뼈대 가 되는 그래프 구조에서 다른 파일의 주소를 통해 접근할 수 있는 형태로 새로이 구성하였다.



<Fig. 5> Graph file system example

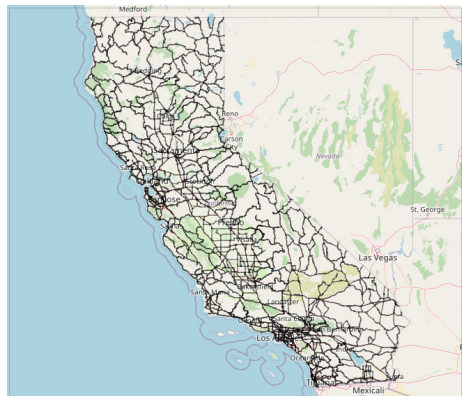
위 <Fig. 5>는 상술한 파일 시스템을 간략히 나타낸 것이다. 기본이 되는 그래프 구조의 노드의 참조 주소를 통해 기하 정보 및 속성 정보로 접근할 수 있는 형태이다. 기본 그래프에는 ID, 연결된 기하 정보로 접근할 수 있는 주소 정보, 속성 정보로 접근하기 위한 주소, 이전 및 이후 연결된 링크에 대한 정보를 포함한다. 이와 같이 피처의 연결성은 해당 파일에 저장되고 변경 시 그래프 구조 자체만 수정하면 빠르게 피처 간 위상관계의 변경점 적용이 가능하다. 그래프의 피처가 변경되어 해당 피처의 기하 및 속성 정보를 수정하려면 파일상의 물리적 주소에 접근하여 해당 정보를 수정하는 식으로 간단히 변경할 수 있다. 기존 지도 갱신이 기하 정보 수정 이후 위상구조 재정립 순으로 진행되는데 비해 해당 방식은 위상구조가 변경되어도 기하정보의 수정이 필수적이지 않으므로 지도의 위상구조 변경이 빠르다는 점과 기하정보에 대한 의존성이 상대적으로 낮아진다는 점에서 차량의 실시간 주행환경을 고려하면 이전의 방식보다는 상대적 강점을 가질 수 있을 것으로 판단된다.

### III. 실험 및 결과

다음은, 작성된 그래프 구조에 대한 실험 내용이다. Python 3.4 버전을 이용하여 실제 도로망 데이터를 인접 리스트 형태로 구현하고, 구현한 인접 리스트를 이용하여 노드의 추가, 변경 및 삭제 작업에 걸리는 시간을 측정하였다. 사용한 자료의 상세는 다음 <Table 1>과 같으며, <Fig. 6>는 OpenStreetMap의 백그라운드 지도에 자료를 오버레이하여 시각화한 것이다.

<Table 1> Example data specification

Area	California, USA	
Data	Node (21,048 rows)	Node ID, X coordinate, Y coordinate
	Edge (21,693 rows)	Edge ID, Start Node, Ende Node, Distance



<Fig. 6> California Road Network

Utah 대학에서 연구용으로 배포하고 있는 미국 캘리포니아 주의 도로망 데이터를 사용하였으며, 간단한 위경도 좌표와 노드, 간선의 ID 및 간선의 길이가 포함되어 있는 자료를 사용하였다. 그래프를 구성하는 노드는, 자료에서 두 개의 좌표를 가진 간선을 노드로 간주하였다.

아래 <Table 2>는 위의 노드, 간선 자료들을 인접 리스트로 나타낸 이후 노드와 간선의 추가 삭제 작업을 수행하고 처리 시간을 나타낸 표이다. 이후의 모든 실험은 Intel i7-7700k cpu, 16gb RAM 환경의 동일한 PC 간 Server-Client 환경을 임의로 구축하고 Client의 request 처리 및 결과 반환까지의 시간을 측정하였다.

<Table 2> Graph data Structure Transaction test results

Scenario	Count	Result(ms)	Scenario	Count	Result(ms)
Add Node	100	0.0010	Add Edge	100	0.0001
	200	0.9989		200	0.0001
	300	1.0002		300	0.9918
	400	1.8189		400	2.0001
Delete Node	100	23.0050	Delete Edge	100	1.0002
	200	34.0080		200	0.9999
	300	48.0099		300	1.0011
	400	61.0142		400	1.9999

총 400회 가량의 테스트에서 노드, 간선에 대한 추가 삭제 작업을 반복적으로 수행하여 누적 처리시간을 측정하였다. 노드 추가 시 임의의 다른 노드와 연결관계를 구성하고, 삭제 시에는 연결관계를 임의로 삭제하는 방식으로 실험을 진행하여 노드 삭제 시 연결관계 삭제에 소요되는 시간이 두드러지게 길게 나타난 것을 확인할 수 있다. 단순 간선의 추가 삭제의 경우 노드의 단순 추가보다는 시간이 약간 더 소모된 것을 확인하였다. 이 과정은 도로망에서의 교차로(노드), 교차로와 연결된 도로(간선)의 추가 삭제에 대응하는 것으로, 차량 단말로 특정 교차로나 도로에서 사고 등의 이유로 실시간 갱신이 필요한 경우 외에도 MMS 장비 또는 차량의 센서를 통해 인식된 실시간 도로 갱신 정보가 서버 또는 다른 차량 단말의 지도 정보를 갱신하는 경우에도 해당 작업을 수행하게 된다. 실제 지도 정보는 이후 연결된 기하 정보와 속성 정보 전체가 갱신되었을 때 업데이트 작업이 완료되나, 실시간 경로탐색 및 계획을 위해 그래프 구조를 선제적으로 갱신하는 경우를 가정하여 그래프 구조를 갱신하는 부분만을 한정적으로 테스트하였다.

위 과정에서 노드의 삭제를 제외한 나머지 경우에서 소요시간이 유사하게 나타났는데, Python 환경에서 Dictionary 형태로 작성된 자료형에서 노드와 연결된 간선의 수가 많지 않은 희소 그래프를 수정할 때 Dictionary 자료형 내부에서 원하는 정보에 접근하기 위한 검색 시간이 크게 차이가 나지 않기 때문으로 추정된다.

추가적으로, 위 표에서 얻은 처리 결과가 일반적인 타일 캐싱 방식의 지도 갱신 방법과 얼마나 속도 차이가 나는지 확인하기 위해 Geoserver를 이용하여 동일한 자료를 레이어로 발행한 이후 로컬에서 WMS(Web Map Service)를 구성하였다. WMS는 이미지 파일을 미리 만들어두고 필요에 따라 렌더링을 위해 타일을 전송하게 되는데, 이 경우 실질적인 공간 객체의 형상 정보 및 위상구조 갱신을 하는 것이 아닌, 단순히 렌더링을 위한 타일 캐싱 방식이다. 논리적으로 정확한 비교를 위해서는 WMS가 아닌 WFS(Web Feature Service)와 같은 기하 정보와 위상 구조 모듈을 포함하는 타일의 전송 방식을 직접 비교하여야 하나, 실질적으로 WFS 전송 테스트는 비교가 의미 없을 정도로 전송과 갱신에 시간이 많이 소모되므로 서버의 캐시 메모리상에 저장된 저용량의 타일 이미지를 전송하는 WMS를 비교하였다.

총 15레벨의 타일을 미리 작성해서 캐시 메모리에 저장하고 실시간 렌더링 시의 타일 전송 시간을 측정하였다. 또한, WMS의 타일 단위 지도 캐싱이 랜덤으로 이루어지는 것을 고려해 갱신할 타일이 고정된 경우를 가정하여 갱신할 자료의 MBR(Minimum Bounding Rectangle)을 계산하고 해당 MBR이 포함된 가장 작은 크기의 타일을 미리 검색한 이후 갱신하는 경우를 가정하고 해당 타일의 갱신 시간을 추가적으로 측정하였다. 다음 <Table 3>은 WMS를 사용한 경우의 타일 전송 시간 및 MBR을 계산한 경우의 타일 전송 시간을 비교한 자료이다.

<Table 3> Tile caching results

Scenario	Count	Result(ms)
WMS Tile Caching	100	4112.1440
	200	7856.2470
	300	12179.8060
	400	14520.3620
MBR Pre-calculate	100	4112.1440
	200	7856.2470
	300	12179.8060
	400	14520.3620

WMS의 경우 가장 작은 15레벨의 타일의 경우 단위가 수십 미터(m)의 범위부터 0레벨은 캘리포니아 주 전체 범위를 전송하는데, 평균 타일 당 전송시간은 15레벨 타일 7ms, 1레벨 타일 133ms로 나타났다. MBR을 미리 계산하여 갱신할 타일을 고정된 경우에도 갱신 타일의 전송시간은 WMS와 유사하게 나타났다. 이는 갱신 자료를 무작위로 생성하는 경우, WMS를 이용하여 갱신 타일을 무작위로 보내는 것과 결과적으로는 동일한 양상을 보이는 것이 이유로 추정된다.

지금까지의 결과를 비교해 보면, WMS의 경우 가장 작은 15레벨의 타일의 경우 단위가 수십 미터(m)의 범위부터 0레벨은 캘리포니아 주 전체 범위를 전송하는데, 평균 타일 당 전송시간은 15레벨 타일 평균 7ms, 1레벨 타일 평균 133ms로 나타났다. MBR을 미리 계산하여 갱신할 타일을 고정된 경우에도 갱신 타일의 전송시간은 WMS와 유사하게 나타났다. 이는 갱신 자료를 무작위로 생성하는 경우, WMS를 이용하여 갱신 타일을 무작위로 보내는 것과 결과적으로는 동일한 양상을 보이는 것이 이유로 추정된다.

비슷한 목적을 가진 기존의 연구들이 지도의 타일을 동적으로 설정하거나, 유희 시간을 통한 지속적 업데이트를 이용하여 업데이트를 진행하므로 이전의 타일이 정해진 형식의 갱신 방식에 비해 속도나 편의성이 증가하였을 수는 있으나 실시간성이 떨어지고, 타일 최신화를 수행하였으나 단말기에 탑재된 지도의 버전 관리가 오히려 어려워지는 등 장·단점이 존재하였다. 그러나 지금까지 수행한 실험을 통해, 갱신할 그래프 부분만을 선제적으로 처리하여 차량의 실시간 주행 상황에서 도로의 폐쇄, 신설, 공사 등으로 인한 일시적 이동 불가와 같은 정보를 선반영하고 경로를 탐색한다는 목적에는 도로망 구조의 변경사항을 우선적으로 적용하고 형태, 좌표와 같은 기하 정보 및 기타 속성정보를 추후에 반영하는 본 논문의 제안방식이 좀 더 효율적일 것으로 판단된다.

#### IV. 결 론

이 연구는 도로망 자료를 그래프 구조로 표시하여 기하 정보에 비 의존적인 형태로 빠르게 갱신하는 방안 에 관해 다루었다. 이를 위해 단일 연결 리스트와 이중 연결 리스트를 활용하여 인접 리스트를 구성하고, 실제 도로 자료를 인접 리스트 형태로 구현하고 노드와 간선 정보를 다루는 방법을 간략히 정의하였으며 소요 시간과 예상 활용 방안을 검토하였다.

실제 차량이 이동하면서 필요한 경로 갱신 정보는 차량 주변의 비교적 작은 범위에 해당한다는 것을 감안 하면 본 연구의 실험에서 수백 개 이상의 변경사항을 처리하는 것 보다는 수십 개 정도의 변경사항을 일시



에 처리하는 것이 일반적인 경우라고 간주할 수 있다. 본 실험의 검토 결과를 살펴보면 사용한 자료가 상당히 넓은 공간적 범위를 가진 자료임에도 불구하고 전체 그래프 구조의 생성과정은 수십 초 이내, 일부 변경은 0.1 마이크로초가 되지 않는 짧은 시간에 이루어지는 것을 알 수 있다. 이러한 장점 때문에, 기존의 기하 정보를 기반으로 한 도로 구조 갱신 방식보다는 경로 정보 전송 또는 특정 형태의 위치 참조를 목적으로 하는 상황에서는 현격히 빠른 도로지도 업데이트가 가능할 것으로 보인다.

위와 같은 도로의 형태를 반영하기 쉬운 간편한 구조와 빠른 속도의 장점이 있기 때문에, 통제된 환경에서의 실험을 수행했다는 한계점을 고려하더라도 충분한 갱신 및 갱신 정보의 전송 속도를 확보할 수 있다면 앞으로 개인의 휴대 기기에서 수집되는 정보, IOT 시스템과 같은 여러 자료 수집처에서 지도 정보를 실시간으로 수집, 갱신하는 목적으로의 확장성을 기대할 수 있다.

하지만 실험 과정에서의 한계로, 위상 구조와 기하 정보 모두에 접근할 수 있는 형태로 전송이나 갱신이 수행되는 그래프 구조 이용 방식과 렌더링을 위한 그림 파일 형태로 갱신을 수행하는 Tile Caching 방식을 비교하였다는 문제점이 있으므로 노드와 간선의 추가 삭제가 단일로 이루어지는 것이 아니라 대규모 갱신일 경우의 복잡한 시나리오로 검증하였을 때 타일 단위 갱신에 이점이 있을 수 있는 부분에 대한 추가적인 실험이 필요하다.

또한, 도로 구조 갱신에 대한 전제로, 송/수신하는 지도의 구조가 동일하다는 가정을 하였다는 한계점이 있다. 따라서 이·기종 간의 지도 정보를 교환하기 위한 위치참조와 같은 작업을 수행할 때 이러한 위상 구조를 이용한 갱신 방안을 활용하려면 서로 다른 좌표계나 다른 플랫폼 간 정보 호환을 적용하기 위한 표준과 같은 자율주행 차량 지도에 대한 연구가 지속적으로 필요하다. 최근 자율주행 상황에서 사용되는 고정밀 지도에서 사용되는 도로의 기본 단위가 차로 수준이라는 것을 감안하면 차로 단위의 도로 구조를 이용하는 방안에 대한 연구 또한 이루어져야 할 것이다. 또한, 그래프에 저장되는 정보가 단순히 도로 중심선을 기반으로 하였다는 점에서 면 형태의 실제 도로 표현에는 물리적 한계성이 존재하므로 기하 정보 및 속성정보의 저장을 위한 추가적인 구조의 개선이 필요하다.

그럼에도 불구하고, 기존의 기하 정보를 기반으로 한 갱신 방안보다는 차량의 실시간 주행 환경이라는 특정 상황에서의 더 나은 속도 및 간편성을 보장한다는 점에서 이 연구가 의의를 가지며, 이를 활용한 실시간 대용량 정밀도로지도 갱신 정보 전송 및 갱신 목적에 유용하게 사용될 것으로 기대해 본다.

## ACKNOWLEDGEMENTS

본 연구는 국토교통부 국토교통과학기술진흥원의 국토공간정보연구사업 (과제명 “자율주행 지원을 위한 도로변화 신속 탐지, 갱신 기술 개발 및 실증”, 과제번호: 18NSIP-B145070-01)의 연구비 지원에 의해 수행되었음

## REFERENCES

Beyen J., Ziems M., Mueller S., Roovers S. and Heipke C.(2012), “Semi-automatic update and quality control of road databases,” *Proc. of the 4th Annual Conference, GEOBIA, Rio de Janeiro, Brazil*, pp.443-448.

- Ga C. O., You K. Y., Sim J. B. and Kim H. T.(2008), “The Proposal and Simulation of Path Unit’s Network Data Update Method Using Wireless Network,” *Journal of the Korean Society for Geospatial Information Science*, vol. 9, no. 3, pp.29-34.
- Jomrich F., Sharma A., Rueckelt T., Burgstahler D. and Böhnstedt D.(2017), “Dynamic Map Update Protocol for Highly Automated Driving Vehicles,” *Proc. of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems, VEHTS*, Porto, Portugal, pp.68-78.
- Kim K. D., Lee D. G., Yu Y. G. and Lee H. J.(2017), “Processing of Aerial Photographic Image Using Unmanned Aerial Photogrammetry Process and Revision/Update Method of Digital Map of Small Change Area,” *Journal of the Korean Society for Geospatial Information Science*, vol. 25, no. 4, pp.15-24.
- Real Datasets for Spatial Databases: Road Networks and Points of Interest, <https://www.cs.utah.edu/>, 2020.01.16.
- Seol J. H., Lee W. J., Choi Y. S. and Jeong I. H.(2019), “A Study On the Renewal System of Domestic High Definition Maps,” *Journal of the Korean Association of Geographic Information Studies*, vol. 22, no. 1, pp.133-145.