

An Unified Spatial Index and Visualization Method for the Trajectory and Grid Queries in Internet of Things

Jinju Han*, Chul-Won Na*, Dahee Lee*, Do-Hoon Lee*,
Byung-Won On*, Ryong Lee**, Min-Woo Park**, Sang-Hwan Lee**

Abstract

Recently, a variety of IoT data is collected by attaching geosensors to many vehicles that are on the road. IoT data basically has time and space information and is composed of various data such as temperature, humidity, fine dust, Co2, etc. Although a certain sensor data can be retrieved using time, latitude and longitude, which are keys to the IoT data, advanced search engines for IoT data to handle high-level user queries are still limited. There is also a problem with searching large amounts of IoT data without generating indexes, which wastes a great deal of time through sequential scans. In this paper, we propose a unified spatial index model that handles both grid and trajectory queries using a cell-based space-filling curve method. also it presents a visualization method that helps user grasp intuitively. The Trajectory query is to aggregate the traffic of the trajectory cells passed by taxi on the road searched by the user. The grid query is to find the cells on the road searched by the user and to aggregate the fine dust. Based on the generated spatial index, the user interface quickly summarizes the trajectory and grid queries for specific road and all roads, and proposes a Web-based prototype system that can be analyzed intuitively through road and heat map visualization.

▶ Keyword: IoT(Internet of Things), Spatial Index, Query Processing, Data Visualization, Web System

1. Introduction

사물 인터넷(Internet of Things, IoT) 기술이 발전하면서 점차 다양한 단말들이 사물 인터넷에 연결되고 있다. 그에 따른 단말들이 수집하는 사물 데이터도 급증하는 추세이다. 많은 양의 사물 데이터를 분석하여 사람들에게 유용한 서비스를 제공할 수 있기 때문에 사물 데이터의 활용도는 점점 높아지고 있다. 최근 도로를

따라 주행하는 각종 차량에 RFID, GPS, 온도계, 습도계, 미세먼지 측정기 등의 지오센서(Geosensor)를 부착하여 실시간으로 사물 데이터를 수집한다. 온습도, 미세먼지, 이산화탄소 등의 다양한 사물 데이터는 데이터가 수집되는 시간과 장소(위도 및 경도) 정보와 함께 실시간으로 데이터베이스에 저장된다.

• First Author: Jinju Han, Corresponding Author: Ryong Lee

*Jinju Han (jinjuhan7@kunsan.ac.kr), Dept. of Software Convergence Engineering, Kunsan National University

*Chul-Won Na (ncw0034@kunsan.ac.kr), Dept. of Software Convergence Engineering, Kunsan National University

*Dahee Lee (dahee@kunsan.ac.kr), Dept. of Software Convergence Engineering, Kunsan National University

*Do-Hoon Lee (dhlee7@kunsan.ac.kr), Dept. of Software Convergence Engineering, Kunsan National University

*Byung-Won On (bwon@kunsan.ac.kr), Dept. of Software Convergence Engineering, Kunsan National University

**Ryong Lee (ryonglee@kisti.re.kr), Research Data Sharing Center, Division of National Science and Technology Data, Korea Institute of Science and Technology Information

**Min-Woo Park (pminwoo@kisti.re.kr), Research Data Sharing Center, Division of National Science and Technology Data, Korea Institute of Science and Technology Information

**Sang-Hwan Lee (sanglee@kisti.re.kr), Research Data Sharing Center, Division of National Science and Technology Data, Korea Institute of Science and Technology Information

• Received: 2019. 08. 14, Revised: 2019. 09. 05, Accepted: 2019. 09. 06.

• This work was supported by an R&D project, 'Enabling a System for Sharing and Disseminating Research Data (K-19-L01-C03) of Korea Institute of Science and Technology (KISTI), Korea.

사물인터넷 확산에 기여한 핵심 기술은 단말과 인터넷 연결 및 단말 간 데이터 전송 기술에 그쳤었지만, 최근에는 단순히 사물 연결에 그치지 않고 단말들이 수집하는 사물 데이터를 분석하여 활용하려는 시도가 점차 늘어나고 있다[1]. 하지만 지금까지 사물 데이터에 대한 진전된 검색 기술은 미흡한 실정이다. 기존에는 IoT 센서 선택 후 해당 센서의 현재 또는 과거 데이터를 탐색하거나 지정한 공간 또는 시간 범위 내에 수집된 사물 데이터 검색 등 단순 브라우징 및 단순 검색만 지원한다. 사물 데이터의 키 값인 시간, 위도, 경도를 이용해 간단한 센싱 정보는 검색할 수 있지만 그 외의 사용자 쿼리는 개선이 필요하다. 예를 들어 특정 도로 구간에 대하여 통행량이나 미세먼지 집계 등의 고차원적 분석을 하기 위해서는 단순히 시간, 위도, 경도의 키 값으로 사물데이터 검색이 불가능하다. 또한 많은 양의 사물 데이터에서 색인을 생성하지 않고 검색하면 순차적 스캔을 통해 엄청난 시간과 비용이 소요된다. 따라서 실시간으로 계속 수집되는 방대한 양의 사물 데이터에서 원하는 데이터를 효율적으로 검색하기 위해서는 공간 색인 구축이 필요하다. 본 논문에서는 실시간으로 저장되는 차량 데이터베이스에서 차량이 주행한 도로 네트워크에 대한 사용자의 이질적인 쿼리를 빠르게 처리하기 위하여 단일 공간 색인 모델을 구축한다.

먼저 실제 대구의 택시 18대에 센서 모듈을 설치하여 약 1년간 수집된 사물 데이터의 GPS로 트레젝토리를 생성한다. 본 연구는 이질적인 쿼리를 빠르게 처리할 수 있도록 단일 셀 기반의 공간 채움 곡선(space-filling curve) 방법을 이용해서 도로 및 트레젝토리의 공간 색인을 구축한다. 트레젝토리 쿼리는 사용자가 검색한 도로 구간에 택시가 지나간 트레젝토리의 셀들을 찾아 통행량을 집계한다. 그리드 쿼리는 사용자가 특정 도로에 대해 검색하면 도로의 셀들을 찾고 미세먼지 농도를 집계한다. 구축된 색인을 바탕으로 유저 인터페이스에서 특정 도로와 전체 도로에 대한 트레젝토리 및 그리드 쿼리를 빠르게 요약하고, 도로별 시각화 및 히트맵의 시각적 표현을 통해 직관적으로 파악할 수 있는 웹 기반 프로토타입 시스템을 제안한다.

도로와 택시 사물 데이터의 맵 매칭 및 색인 구축을 통해 각 도로에 대한 통행량 및 미세먼지 농도를 집계할 수 있다. 또한 도로 및 히트맵 시각화를 통해 검색 날짜, 시간별로 도로의 통행량과 미세먼지 농도를 직관적으로 파악할 수 있다.

본 연구의 기여도는 다음과 같다.

- 기존 연구와 비교하였을 때 본 연구는 사물 데이터를 보다 고차원적으로 검색하고 분석할 수 있는 독창적인 방법이다.
- 사용자의 이질적인 쿼리(트레젝토리와 그리드 쿼리)들을 처리하기 위해서 셀 기반 공간 채움 곡선 방법의 단일 공간 색인을 구축하여 빠르게 검색한다.
- 사용자 쿼리를 요청 받으면 단순히 해당 센서 데이터들을 반환하는 것이 아닌 해당 그리드 및 트레젝토리를 셀 단위로 요약하여 집계한다.
- 특정 도로와 전체 도로의 혼잡도 및 미세먼지 쿼리를 빠르게 처리할 수 있는 웹 유저 인터페이스를 구축하고, 결과를

직관적으로 분석할 수 있는 시각화 시스템을 구축하여 웹 기반 프로토타입 시스템을 제안한다.

- 본 연구를 통해 시간별로 도로의 혼잡도 및 미세먼지를 파악하여 차량 정체 구간 및 미세먼지 경보 등 유용한 알람 서비스를 제공하는데 도움을 줄 수 있다.

이 논문의 구성은 다음과 같다. 제 2장에서는 제안 방안의 이해를 돕기 위해 관련 연구에 대하여 기술한다. 그리고 제 3장에서는 제안 방안에 대해 설명한다. 제4장에서는 제안 방안에 대한 실험 결과에 대하여 기술한다. 마지막으로 제 5장에서는 결론 및 향후 연구 방향을 다룬다.

II. Related works

이 장에서는 제안하는 기법의 이해를 돕기 위해서 본 연구와 비슷한 문제를 해결하는 관련 연구를 기술한다. 또한 기존의 연구와 본 연구의 차이점을 기술한다.

센서 데이터 등의 기본적인 스트림 처리를 위해서 B-Tree를 활용한 1차원 질의처리 구조가 대표적으로 활용된다. 예를 들어 다수의 센서 정보를 가지고 질의를 구성한다면 1차원 질의 처리 구조는 모든 센서 정보의 질의를 한 곳에 관리하는 것이 아니라 온도질의, 습도질의 등과 같이 나누어서 구성하는데 사용된다. B-Tree 구조는 사용자가 등록한 질의가 트리의 각 단말 노드에 저장되며, 질의범위의 시작 값에 따라 정렬되어 구성된다. 따라서 트리에 저장되어 있는 질의를 탐색하기 위해서는 최상위 노드부터 마지막 노드까지 트리의 높이만큼 탐색하여 효율적인 탐색 경로를 제공한다. 하지만 트리에 담겨진 질의 정보는 단일 데이터로 정해져 있어 범위 질의에 대한 질의는 등록할 수 없다. 이에 1차원 질의처리 구조는 다양한 질의를 표현할 수 없기에 서비스 환경에 적용하기 힘든 구조임을 알 수 있다[2].

[3]은 각각의 속성에 대하여 B+ Tree를 만들어, 어떠한 속성으로 검색이 되더라도 유사한 센싱 데이터 집합을 빠르게 검색할 수 있도록 한다. 하지만 B+ Tree는 대용량 데이터 저장의 경우 과도한 제자리 갱신(in-place update) 비용이 매우 크다. 대용량 공간 데이터를 빠르게 검색하기 위한 색인 모델을 개발할 필요가 있다. [4]는 RQMAT(Range Queries on Multi-attribute Trajectories)에 효율적으로 응답하기 위해 다중 속성 궤도(Multi-attribute Trajectory)를 관리하는 R-Tree 색인 구조를 제안한다. 하지만 R-Tree는 데이터가 계속 변경되면 경계 상자(Bounding Box)에 과도한 겹침이 생겨 트리의 노드를 분리하고 재조정해야 하기 때문에 유지보수 비용이 든다. [5]에서 R-tree 기반의 질의처리 기술은 입력 데이터가 중복된 질의 영역에 있다면 트리의 높이만큼 탐색하는 것이 아니라 만족되는 질의가 끝날 때까지 트리의 노드를 계속 탐색해야 한다. 수많은 센서 데이터는 매우 다양한 범위 질의가 등록되기 때문에 대부분 질의가 중복된다. 결국 각 이벤트에 대해 트리

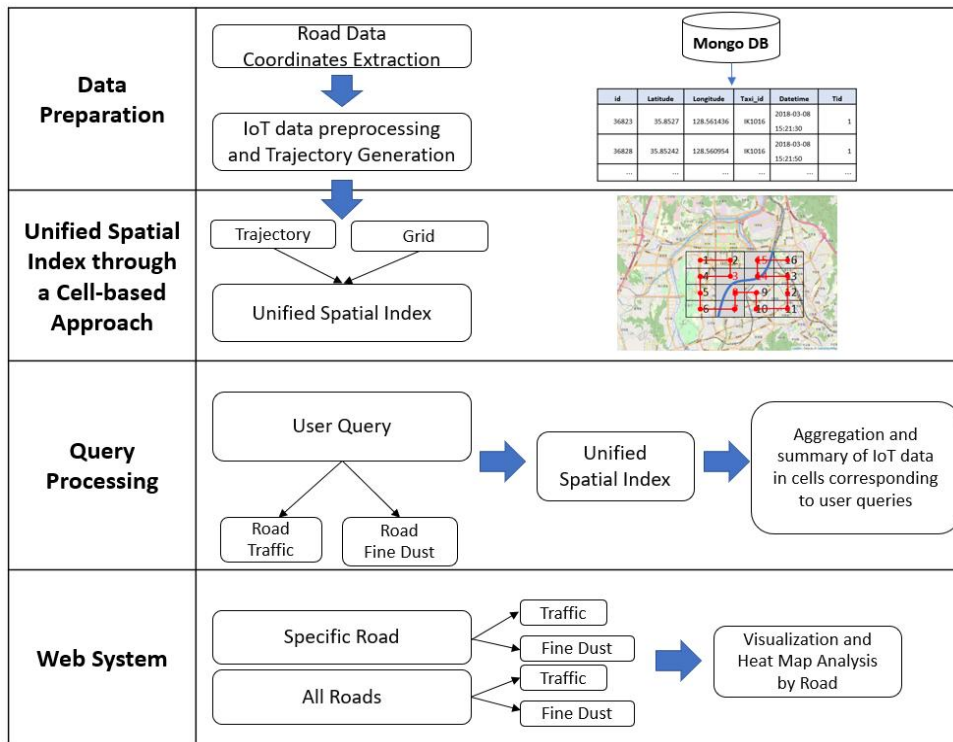


Fig. 1. Flow Chart of the Proposed Model

전체에 동일한 탐색과정을 반복해야하므로, 이를 효율적으로 감당할 수 없다.

[6]은 택시의 대용량 센서 데이터를 이용하여 택시의 GPS 트래젝토리를 구축하여 택시의 이동 시간을 예측한다. 하지만 도시의 도로 영역(nearest neighbor) 및 택시 사물 데이터 집계 쿼리 처리 및 시각적 분석 기법이 부재하다. [7]의 데이터 큐브는 대용량 데이터를 사전 계산하여 집계한다. 하지만 관계형 데이터베이스에서 색인 없이 집계 쿼리를 사전 계산하여 엄청난 양의 디스크 공간을 차지한다. 또한 지속적으로 수집되는 사물 데이터에 대한 요약 또는 빈번히 사용되는 쿼리에 대한 요약이 수행되지 않아 처리시간이 길어지고 실시간 분석에 어려움이 예상된다.

[8]은 뉴욕시에서 2010년부터 2013년 사이에 발생한 70억 개의 택시 데이터를 통계적 기법으로 분석하여 뉴욕 시의 교통 흐름을 파악하였다. 또한 지도 위의 경로마다 교통량을 표현하여 시각적 효과를 높였다. [9]는 택시 데이터의 GPS를 이용하여 Time series에 따라 동적으로 시각화하였다. 그 결과 도시의 교통 흐름을 시각적으로 인지할 수 있었고 데이터의 범위가 확장되는 경우 도시와 도시간의 교통 흐름도 알 수 있었다. 하지만 [8,9]는 택시의 GPS의 위도, 경도의 데이터만 활용하여 도시의 교통량을 특정 도로 영역이 아닌 단순히 트래젝토리를 구축하여 시각화하였다.

센서 네트워크에서 한 사용자가 질의를 여러 개 보내거나 여러 사용자로부터 다양한 질의가 센서 네트워크에 보내어질 때 이를 처리하는 간단한 방법으로는 감지한 센서 데이터를 각각의 질의에 관련이 되는지를 순차적으로 비교하는 것이다. 그러

나 이 방법의 경우 모든 질의들에 대해 모든 센서 데이터를 하나하나 다 비교해야 하므로 센서 데이터 하나당 질의의 수만큼 비교를 해야 한다[10].

[8,9,10]은 사물 데이터에 대한 사용자 쿼리를 빠르게 처리할 수 있는 색인이 없다. 또한 기존에는 단일한 색인이 없기 때문에 이질적인 사용자 쿼리를 처리하려면 각각의 색인을 구축해야 한다. 본 연구는 이질적인 데이터를 셀 기반의 단일 색인에 적용하여 택시 사물 데이터를 도로 영역에 매칭 시키고, 트래젝토리 및 도로 영역의 셀에서 통행량 및 미세먼지를 집계한다. 따라서 단일한 셀 기반의 공간 채움 곡선 방법을 이용하여 색인을 구축하고, 트래젝토리와 그리드 쿼리의 이질적인 쿼리에 빠르게 응답할 수 있다. 또한 도시의 통행량 및 미세먼지 농도를 직관적으로 파악할 수 있는 웹 기반 프로토타입 시스템을 구축하여 도로별 시각화 및 히트맵을 통해 시각적인 분석을 지원한다. 이를 통해 도로의 통행량 및 미세먼지를 파악하여 차량 정체 구간 및 미세먼지 경보 등의 알림 서비스에 도움을 줄 수 있다.

III. The Proposed Scheme

Fig. 1은 제안 방안의 흐름도이다. 대도시를 운행하는 택시들로부터 수집된 데이터에서 언제 어느 택시가 어느 도로를 지나갔는지 알아내기 위해서는 도로 데이터가 필요하다. 먼저 도로 데이터에 좌표 정보가 없기 때문에 도로 구간의 좌표를 추출하였다. 대도시

택시에서 수집된 사물 데이터는 약 0.1초마다 저장되었기 때문에 대용량 데이터의 전처리 과정을 거친 후 택시의 GPS 트래젝토리를 구축하였다. 본 연구는 대구시를 그리드 셀 공간으로 나누고, 공간 채움 곡선(space-filling curve) 방법인 힐버트 곡선(Hilbert curve)을 이용해서 도로 및 차량 트래젝토리의 공간 색인을 생성한다. 또한 도로별 통행량과 미세먼지의 셀 집계를 하기 위한 쿼리 프로세싱을 수행한다. 마지막으로 웹 기반 프로토타입 시스템의 유저 인터페이스를 구축하여 특정 도로와 전체 도로에 대한 통행량 및 미세 먼지 농도에 대한 쿼리를 할 수 있고, 도로의 혼잡도와 미세 먼지 농도별로 색을 다르게 하여 직관적으로 파악하고 분석할 수 있는 시각화 시스템 개발을 목표로 한다.

1. Data Preparation

1.1 Road Data Coordinates Extraction

대구시를 운행하는 택시들로부터 수집된 데이터에서 언제 어느 택시가 어느 도로를 지나갔는지 알아내기 위해서는 대구시 택시 데이터와 도로 데이터가 필요하다. 도로 데이터는 국가 공간정보포털에서 행정안전부가 (도로명주소)도로구간 데이터를 셰이프(shape) 파일 형태로 제공한다[11].

Table Name		Table Definition						
Table Description		Table ID	Z_KAIS_TL_SPRD_MANAGE					
No	Column ID	Column Name	Type	Byte	NULL	PK/FK	Default	Reference Table Name
1	XGEOMETRY	Space image information						
2	SIG_CD (PK)	City and county code	VARCHAR2	5		PK		
3	RDS_MAN_NO (PK)	Road section serial number	NUMBER	12		PK		
4	RN	Road name	VARCHAR2	80				
5	RN_CD	Road name code	VARCHAR2	7				
6	ENG_RN	English road name	VARCHAR2	80				
7	NTRC_DE	Notice date	VARCHAR2	8				
8	WDR_RD_CD	Wide area road classification code	VARCHAR2	10				
9	ROA_CLS_SE	Road hierarchy function classification	VARCHAR2	2				
10	RDS_DPN_SE	Road section type classification	VARCHAR2	1				
11	RBP_CN	the starting point	VARCHAR2	80				
12	REP_CN	the last stop	VARCHAR2	80				
13	ROAD_BT	Road width	NUMBER					
14	ROAD_LT	Roadway Road	NUMBER					
15	BSI_INT	Basic interval	VARCHAR2	5				
16	ALWNC_RESN	Reason of Buyeo	VARCHAR2	254				
17	ALWNC_DE	Grant date	VARCHAR2	8				
18	MVM_RES_CD	Transfer reason code	VARCHAR2	10				
19	MVMN_RESN	Reasons for moving	VARCHAR2	254				
20	MVMN_DE	Transfer date	VARCHAR2	8				
21	OPERT_DE	Operation date	VARCHAR2	14				

Fig. 2. Road table definition

(도로명주소)도로구간 데이터를 다운로드 받아 테이블 정의서를 보면 주요 컬럼은 도로명과 도로명코드, 기점, 종점이 있다. 하지만 도로의 좌표는 기점과 종점에 대한 정보만 있기 때문에 각 도로 구간에 대한 좌표 추출 작업이 필요하다.

QGIS 프로그램이란 데이터 뷰, 편집, 분석을 제공하는 크로스 플랫폼 자유-오픈 소스 데스크톱 지리 정보 체계 응용 프로그램이다 [12]. QGIS 프로그램 사용하여 Fig. 3처럼 벡터-지오메트리 도구-노드 추출을 클릭하면 모든 도로 구간 사이의 노드들을 추출할 수 있고 벡터-지오메트리 도구-지오메트리 컬럼 내보내기/추가를 클릭하면 노드에 대한 좌표를 추출할 수 있다.

하지만 추출된 좌표는 택시 사물 데이터의 GRS80 위경도 좌표계와는 다른 좌표계이기 때문에 비교할 수 없다. 따라서 R 프로그램을 작성해서 현재 지구를 잘 나타내고 있는 타원체 좌표계인 GRS80 위경도 좌표계로 변환시켰다. 최종적으로 각 도로구간 사이의 위경도 좌표를 추출하여 택시 사물 데이터와 비교할 수 있도록 하였다.

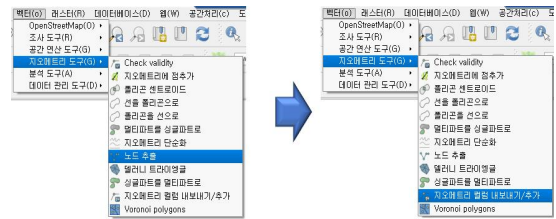


Fig. 3. QGIS Program for Coordinates Extraction

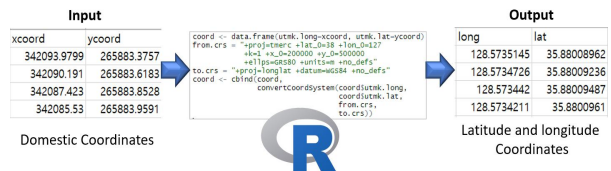


Fig. 4. Coordinate Extraction Process

이때 추출된 좌표에서 중복되는 좌표가 있어 하나의 좌표로 통합을 하였다. Table 1은 중복 좌표 통합 및 좌표 추출 과정을 거쳐 구축한 대구시 도로 데이터의 예시이다. 필드는 도로명 (road), 도로명코드(Road ID), 위도(latitude), 경도(longtitude) 좌표로 구성되어 있다. Fig. 5는 구축된 도로 데이터 예시를 시각적으로 보여주기 위해 공원순환로의 좌표를 오픈스트리트맵 (Openstreetmap)에 표현하였다. 공원순환로에 대해 맵에 추출된 225개의 좌표를 파란색의 폴리라인(polyline)으로 표시하고, 그 중에 15개의 좌표만 샘플링을 해서 빨간색의 마커(marker)로 표시하였다. 따라서 대구시 도로 데이터의 좌표가 구축된 것을 확인할 수 있다.

Table 1. Daegu Road Data Example

Road	RID	Latitude	Longitude
Gongwonsunhwan-ro	3147002	35.8527	128.561436
Gongwonsunhwan-ro	3147002	35.85242	128.560954
...



Fig. 5. Gongwonsunhwan-ro's Coordinate Example

1.2 IoT(Internet of Things) Data of Taxi

Fig. 6는 택시 사물 데이터의 전처리 과정을 흐름도로 나타내었다. 먼저 시공간(시간, 위도, 경도) 정보가 없는 데이터는 제외시켰

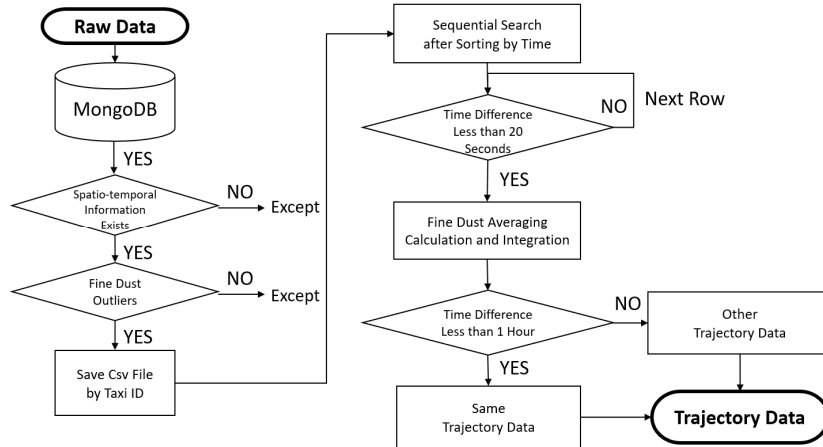


Fig. 6. The Flowchart of Taxidata Preprocessing

다. 미세먼지 농도 기준에 따르면 음수 값은 이상치 값이기 때문에 제외하였다. 택시 아이디별로 파일을 나누어 시간 순으로 정렬한 뒤, 택시마다 수집된 시간 간격이 다르기 때문에 수집된 시간 간격이 가장 긴 20초로 통일시키고 미세먼지 값을 평균 내어 통합하였다. 예를 들어 5초 단위로 데이터가 수집되었으면 12:00:00부터 12:00:19 까지의 센서 값은 평균을 내고 시간, 위도, 경도는 처음에 측정된 값으로 저장하였다. 택시별로 오랜 시간동안 같은 위도, 경도가 수집된 경우는 택시가 정차해있는 것임으로 처음 수집된 시간기준으로 이동하기 전까지 데이터를 통합하였다. 마지막으로 데이터를 순차 탐색하며 첫행과 시간차가 1시간 미만인 데이터들은 같은 택시 궤도로 구축하여 같은 TID(Trajectory ID)를 부여하였다. 이 과정을 반복하여 전체 궤도의 TID를 구축할 수 있다.

Table 2. Daegu Taxi Data Example

ID	Latitude	Longitude	Taxi_ID	PM10	Datetime	TID
36823	35.8527	128.561436	IK1016	22	2018-03-08 15:21:30	1
36828	35.85242	128.560954	IK1016	19	2018-03-08 15:21:50	1
...

Table 2는 택시 사물 데이터의 전처리 과정을 거친 대구 택시 데이터의 예시이다. 이때 ID는 센서의 ID를 나타내고, Latitude는 위도, Longitude는 경도, PM10은 입자 크기가 $10\mu m$ 이하의 미세먼지, Taxi_ID는 택시의 ID, Datetime은 날짜와 시간, TID는 구축된 Trajectory에 대한 ID를 나타낸다.

2. Proposed Method

2.1 Data Model

본 연구에서 제안한 데이터 모델은 이질적이고 다양한 질의들을 통합할 수 있는 기본 요소는 셀(cell)이라고 할 수 있다. 2차원 공간에서 셀은 정사각형으로 표현된다. 실험을 통해 적당히 작은 크기의 셀들을 제안한 셀 기반 색인 모델의 최소 단위로 규정한다. 따라서 그리드 형태의 질의는 셀로 표현할 수 있고, 각 셀은 동일한 크기를 갖는다. 하나의 셀은 하나의 그리드 공간이며, 그 셀이 커버

하는 대구시의 특정 공간을 매핑 한다.

Google s2 라이브러리를 사용하여 각 셀에 대한 셀ID와 그 셀의 네 개의 꼭지점 $\{(a_x, a_y), (b_x, b_y), (c_x, c_y), (d_x, d_y)\}$ 에 대한 좌표를 구할 수 있다. 본 논문에서 대구시 전체 그리드 공간은 연속적인 셀들로 나누어진다. 이때 분할된 각 셀들에 대한 셀ID와 그 셀의 네 개의 꼭지점 좌표들의 공간 정보를 구할 수 있다. 따라서 도로 좌표와 트레젝토리 좌표를 대구의 셀 공간에 매칭시킨다.

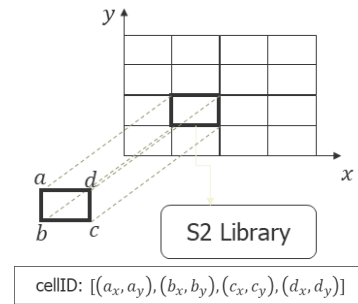


Fig. 7. Cell-based Approach

2.2 Unified Index Model

Fig. 8에서 전체 공간은 4개의 공간으로 분할되며, 전체 공간은 root로, 4개의 분할된 공간은 자식 노드인 00, 01, 10, 11로 표현할 수 있다. 다음 단계에서 각 자식 노드는 4개의 자식 노드들로 분할될 수 있으며, 00에 해당하는 공간은 0000, 0001, 0010, 0011을 레이블로 가진 공간으로 분할된다. 이 때 이들 공간의 접두어(prefix)는 00으로 부모 노드의 레이블과 같다. 다음 단계에서 0010으로 레이블 된 공간은 1000, 1001, 1010, 1011 등 4개의 공간으로 분할된다. 이 때 4개의 공간의 접두어는 10으로 부모노드의 10과 같다. 이것은 전체 공간이 4개의 공간으로 분할되는 과정을 트리로 표현할 수 있고, 이러한 2D Hilbert-curve 패턴은 다음과 같은 순환 함수를 사용하여 구현할 수 있다. 이처럼 2진수의 셀ID는 4진수로 표현 가능하다.

$$\begin{aligned}
 - H_{i+1}(n_0) &= H_i(n) \cdot 00 \\
 - H_{i+1}(n_1) &= H_i(n) \cdot 01
 \end{aligned}$$

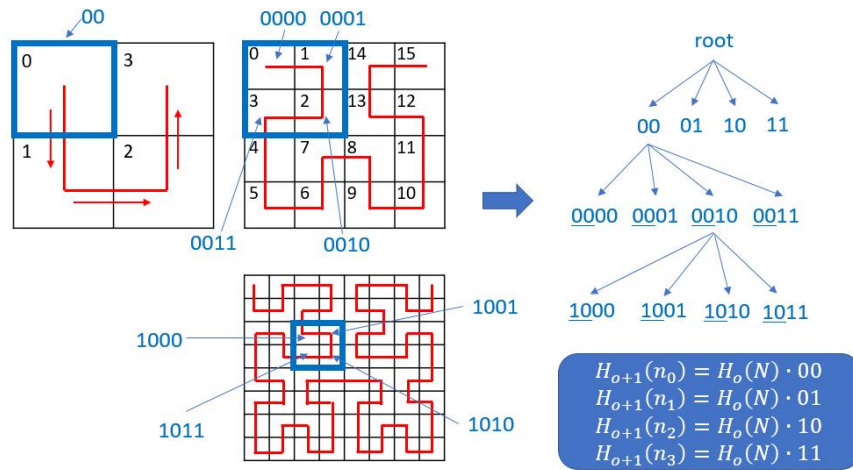


Fig. 8. Implementation of 2D Hilbert Curve

- $H_{i+1}(n_2) = H_i(n) \cdot 10$
- $H_{i+1}(n_3) = H_i(n) \cdot 11$

대도시 공간에 존재하는 셀들은 지역성(locality)과 유사성(proximity)에 의하여 배치된다. 따라서 공간이 유사한 데이터들은 유사한 셀 공간에 존재한다. 즉 어떤 셀과 인접한 셀들은 공간의 유사성이 있다고 할 수 있다. 이러한 공간 유사성에 따라 사물데이터를 저장하기 위해서 본 연구는 2D Hilbert-curve를 이용하여 셀들을 연결하게 된다.

Fig. 9는 4개의 셀, 16개의 셀, 64개의 셀의 중앙점들을 2D Hilbert-curve로 연결한 것을 나타낸다. 2D Hilbert-curve의 특징은 2차원의 공간을 Hilbert-curve를 통해 1차원으로 변환하여 빠르게 탐색할 수 있다. 또한 Hilbert-curve는 공간 채움 곡선들 중에 특히 지역성(locality)을 보존해주는 특징이 있어 다양한 쿼리에 빠르게 응답할 수 있다[13].

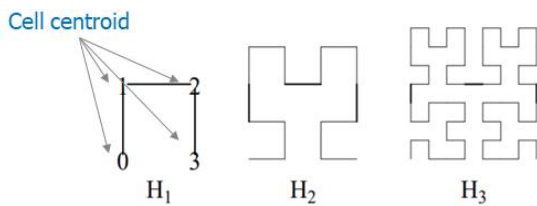


Fig. 9. 2D Hilbert Curve

도로에 대한 쿼리를 처리하기 위해 2D Hilbert-curve를 활용하여 각 도로가 커버하는 셀 정보를 담고 있는 도로 색인을 구축한다. 도로의 통행량 쿼리 프로세싱을 위해 트래젝토리 좌표가 속해있는 셀을 탐지하고 Taxi_ID와 TID(Trajectory ID)로 트래젝토리 색인을 구축하고, 도로의 미세먼지 쿼리 프로세싱을 위해 그리드 셀에 있는 미세먼지 농도를 집계하여 그리드 색인을 구축한다.

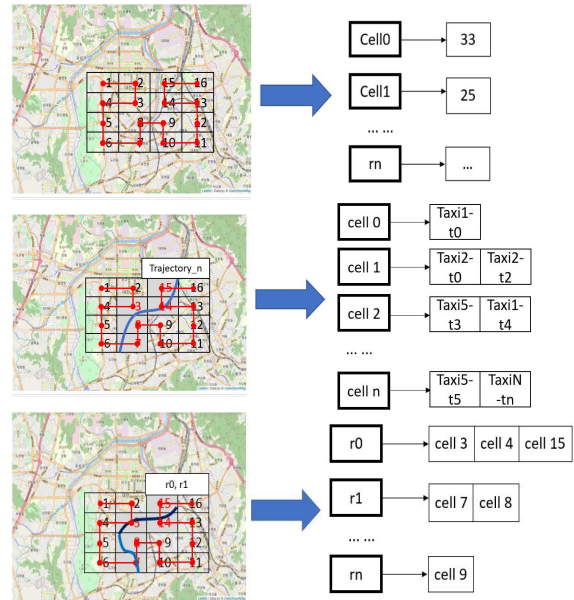


Fig. 10. Cell based Index

2.3 Query Processing

사물데이터의 대한 검색 질의의 종류는 매우 다양하다. 본 연구는 특정 도로와 전체 도로를 기준으로 도로의 통행량 및 미세먼지 쿼리를 지원한다. 본 연구에서 지원하는 쿼리 프로세싱은 다음과 같다.

(1) Road Traffic Query Processing

먼저 통행량 쿼리에 대해 기술한다. 앞에서 전처리한 데이터의 공간 색인 모델을 통해 대도시 주요 도로의 통행량 쿼리 프로세싱을 지원할 수 있다. 특정 도로 검색을 위해 시간, 도로를 입력으로 받아 검색된 시간에 해당 도로를 지나간 트래젝토리를 카운팅한다. 예를 들어 ‘2017-06-03 14:00 ~ 2017-06-03 18:00 대학로에 대한 혼잡도는?’이라는 쿼리가 들어오면 대학로 r0에 해당하는 셀을 빠르게 찾아 트래젝토리 색인에서 각 셀의 Taxi_ID와 TID를 통해 해당 시간의 트래젝토리를 카운팅 해준다.

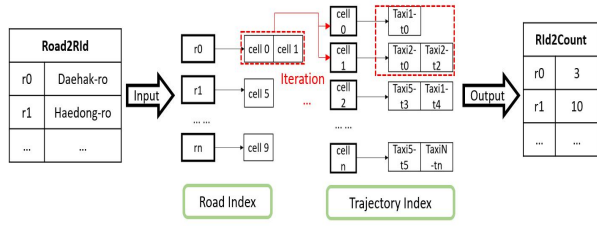


Fig. 11. Specific Road Traffic Flow Query Processing

전체 도로 쿼리는 시작 시간, 끝 시간만 입력으로 받는다. 예를 들어 ‘2017-06-03 14:00 ~ 2017-06-04 01:00 전체 도로에 대한 혼잡도는?’ 이라는 쿼리가 들어오면 해당 시간의 전체 도로에 해당하는 셀을 빠르게 찾아 각 도로별 트래젝토리를 카운팅 해준다.

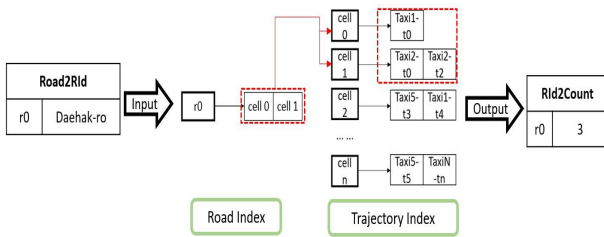


Fig. 12. All Roads Traffic Flow Query Processing

하지만 도로의 통행량 쿼리시 Fig. 13처럼 하나의 셀에 여러 개의 도로가 있는 경우에는 트래젝토리가 지나간 셀의 모든 도로가 카운팅 되는 문제점이 있다.

$$\|p - q\| = \sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2} = \sqrt{\|p\|^2 + \|q\|^2 - 2p \cdot q}$$

$$p = (p_1, p_2, \dots, p_n), q = (q_1, q_2, \dots, q_n) \quad (1)$$

이때 트래젝토리 좌표는 같은 셀에 있는 각 도로 r1, r2 의 모든 좌표와 유클리드 거리 식(1)을 통해 계산한다[14]. 각 도로별로 유사도 평균을 구하여 트래젝토리의 좌표당 가장 가까운 도로를 매칭시킬 수 있다. 따라서 쿼리가 들어오면 해당 시간의 트래젝토리가 지나간 도로만을 정확히 카운팅 한다.

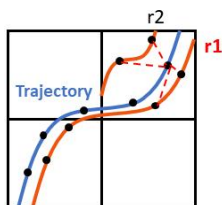


Fig. 13. Multiple Roads in One Cell

(2) Road Fine Dust Query Processing

다음은 미세먼지 농도 쿼리에 대해 기술한다. 앞에서 전처리 한 데이터의 공간 색인 모델을 통해 대규모 주요 도로의 미세먼지(PM10)의 농도를 집계하는 쿼리 프로세싱을 지원한다.

Fig. 14는 특정 도로 쿼리에 대한 쿼리 프로세싱을 도식화한다. 예를 들어 ‘2017-03-08 13:00 ~ 2017-03-08 15:00 대학로에 미세먼지 평균 농도는?’이라는 사용자 쿼리가 들어오면 대학로 r0에 해당하는 셀을 찾아 그리드 색인에서 해당하는 시간의 각 셀별로 미세먼지 농도를 집계한다.

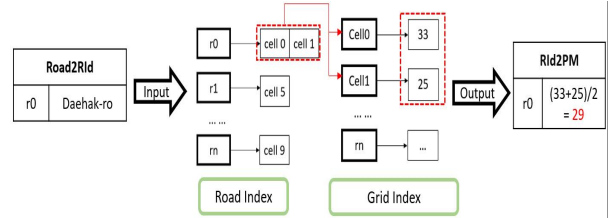


Fig. 14. Specific Road PM10 Query Processing

Fig. 15은 전체 도로 쿼리에 대한 쿼리 프로세싱을 도식화한 것이다. 전체 도로 쿼리는 검색 시간을 입력으로 받아 대규모 전체 도로의 미세먼지를 쿼리한다. 예를 들어 ‘2018-03-08 13:00 ~ 2018-03-08 15:00 전체 도로의 평균 농도는?’이라는 사용자 쿼리가 들어오면 각각의 도로맵에 해당하는 셀을 찾아 그리드 색인에서 각 셀별로 해당하는 미세먼지 농도를 집계한다.

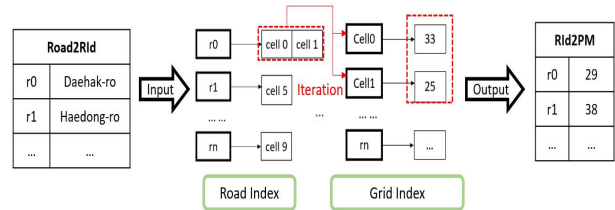


Fig. 15. All Roads PM10 Query Processing

IV. Experimental Results

이 장에서는 실험 환경에 대해 서술하고, 실험 결과인 쿼리 응답 시간 및 웹 기반 프로토타입 시스템에 대하여 서술한다.

1. Experiment Environment

1.1 System Specifications and Software Specifications

실험을 위한 실제 서버 구축 시스템 사양은 Table 3과 같다.

Table 3. System Specifications

System Specifications		Purpose
CPU	Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz	Server construction
RAM	23GB	
HDD	1TB	

Table 4. Software Specifications

Software Specifications		Library Version
OS	Ubuntu 16.04 64bit LTS	
MongoDB	MongoDB 2.6.10	
Python	Python 3.5.2	Pandas 0.24.2 Pymongo 3.8.0 S2sphere 0.25 Folium 0.9.1 Flask 1.0.2
HTML, Javascript, CSS	Htm15 1.0.1 Jinja2 2.10	jQuery 1.11.3

쿼리 프로세싱 및 웹 인터페이스 구축을 위한 소프트웨어 사양은 Table 4와 같다. 본 연구는 파이썬(Python)을 사용하여 색인 모델과 집계 및 시각화를 구현한다. 전처리가 되어있지 않은 대용량 택시 데이터가 저장된 MongoDB에서 약 1년의 센싱 데이터를 Pymongo를 통해 추출하여 전처리 후에 사용한다. 또한 고성능의 데이터구조인 Pandas와 Google에서 개발한 S2sphere 를 사용하여 그리드 셀을 생성한다.

또한 HTML, JavaScript, jQuery, CSS를 사용하여 웹 기반 프로토타입 시스템의 유저 인터페이스를 구축한다. 색인 결과로 도로의 혼잡 정도를 나타내기 위하여 각 도로의 좌표(Coordinates)를 이은 후 혼잡 정도 별로 색을 다르게 시각화하여 나타낸다. 또한 대구시의 미세먼지 농도를 직관적으로 파악하기 위해 도로가 속한 그리드 셀별로 색을 다르게 표시하여 히트맵 분석을 할 수 있다. 이 때 시각화를 위해 Folium을 사용한다. 마지막으로 색인 모델과 웹 기반의 유저 인터페이스가 데이터를 주고 받기 위해 파이썬(Python) 기반 웹 프레임 워크인 Flask를 사용한다.

1.2 Data Statistics

본 연구는 s2 Index의 특정 레벨을 사용하여 대구시의 공간을 연속적인 셀들로 나누었다. Table 5는 셀 레벨별 통계치 데이터이다. 셀 레벨은 0Level부터 30Level까지 있다. 최대 0Level의 면적은 85011012.19km²이고, 셀 레벨의 최소 30Level의 면적은 0.74cm²이다. 본 연구는 셀의 최소 길이가 약 100m정도 되는 16Level의 셀을 사용하였다.

대구시를 운행하는 택시들로부터 수집된 데이터에서 언제 어느 택시가 어느 도로를 지나갔는지 알아내기 위해서는 대구시 도로에 대한 정보가 필요하다.

Table 5. S2 Index Cell Level

	Average area	Min, Max Length	Cell Number
14Level	0.32km ²	425m, 593m	6936
15Level	79172.67m ²	212m, 296m	27,743
16Level	19793.17m²	106m, 148m	110,974
17Level	4948.29m ²	53m, 74m	443,896
18Level	1237.07m ²	27m, 37m	1,775,584
19Level	309.27m ²	13m, 19m	7,102,336

하지만 앞에서 언급한 것처럼 도로 데이터는 좌표에 대한 정보가 부족하기 때문에 QGIS 프로그램과 R 프로그래밍을 통해 위경도 좌표 추출을 하였다. Table 6은 대구시 도로 좌표가 추출된 도로 데이터의 통계치이다. 도로명 개수는 4,007개이고, 그 안에 있는 전체 도로 좌표 개수는 251,124개 이다.

Table 6. Road Data Statistics

Number of roadName	Number of All road coordinates
4,007	251,124

Table7은 본 연구에서 실험을 위해 사용한 택시 데이터 통계이다. 대구 택시 데이터는 총 18대의 택시로 기간은 2017-06-03 14:00 ~ 2018-07-19 12:00 까지 약 1년동안 수집된 센싱 데이터를 사용하였다. 칼럼의 개수는 7개이고, 약 10,000,000개의 센싱 데이터를 사용하였다.

Table 7. Taxi Data Statistics

Number of columns	Number of rows	Number of taxi_ids
7	10,024,307	18

2. Experimental Results

본 연구는 웹 기반 프로토타입 시스템의 유저 인터페이스를 구축하여 특정 도로와 전체 도로에 대한 통행량 쿼리와 미세먼지 집계 쿼리를 빠르게 처리한다. 결과적으로 각 도로의 좌표(Coordinates)를 이은 후 혼잡 정도에 따라 색을 다르게 시각화하고, 미세먼지 농도 별로 도로가 해당하는 그리드 셀 별로 색을 다르게 시각화하여 도로의 혼잡도 및 미세먼지를 직관적으로 파악하고 결과를 히트맵 분석 할 수 있다.

2.1 Query Response Time Comparison

먼저 제안 방안의 작업 프로세스 별 수행시간은 Fig. 16과 같다. 이처럼 셀 기반의 색인을 한번 구축해놓으면 도로에 대한 통행량 및 미세먼지 쿼리를 빠르게 처리할 수 있다.

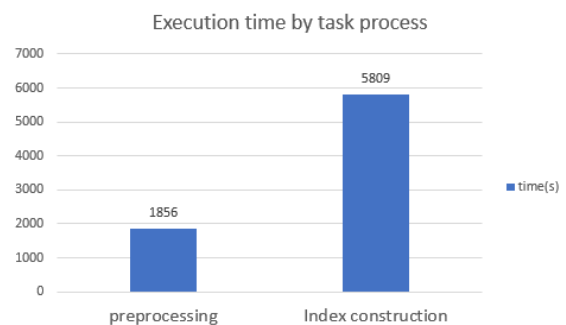


Fig. 16. Execution Time by Task Process

본 연구의 제안 한 셀 기반의 색인을 구축하였을 때 사용자 쿼리 응답시간과 색인을 구축하지 않았을 때 사용자 쿼리 응답

시간을 비교하였다. 색인이 구축되지 않았을 때에는 도로와 트레젝토리가 매칭되지 않아 각 도로의 좌표들과 트레젝토리의 유사도 계산을 통하여 맵 매칭시킨 후에 도로의 통행량 및 미세면지 농도 집계 쿼리를 처리 하였다. 베이스라인은 색인을 구축하지 않고 쿼리 처리를 한 경우이다. Fig. 17은 특정도로 실험 결과 제안 방안의 쿼리 응답시간은 약 0.5 밀리초가 소요되고, 색인을 생성하지 않았을 때엔 약 18 밀리초가 소요된다.

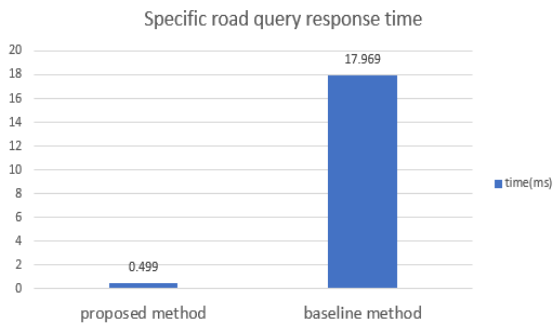


Fig. 17. Specific Road Query Response Time

Fig. 18은 전체도로 실험 결과 제안 방안의 쿼리 응답시간은 2초가 소요되고, 색인을 생성하지 않았을 때엔 72초가 소요된다. 따라서 본 연구의 제안 방안이 약 36배 빠르게 쿼리를 응답할 수 있었다. 이는 하루치에 대한 쿼리를 실행하였을 때 쿼리 응답시간이다. 쿼리 검색 기간을 늘린다면 쿼리 시간은 조금 더 소요될 것이다.

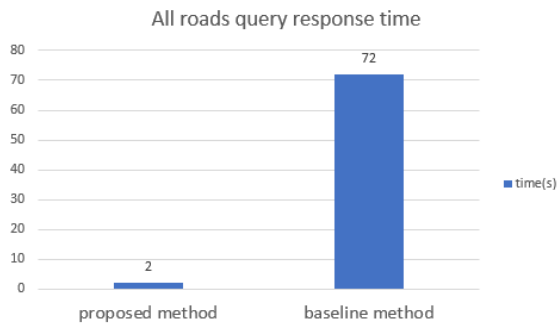


Fig. 18. All Roads Query Response Time

2.2 Road Traffic in Daegu

Fig. 19는 구현한 유저 인터페이스의 대구시 도로 통행량 웹 페이지 소개(Manual)에 대한 화면이다. 매뉴얼에는 사용자 쿼리를 하기 위해 입력 값, 출력 값, 결과 값 등의 기본적인 설명을 기술하였다. “Search 메뉴” 버튼을 클릭하면 특정도로 및 전체도로 검색을 위한 페이지로 넘어간다. 검색결과로 도로의 혼잡 정도를 나타내기 위하여 각 도로의 좌표(coordinates)를 이은 후 혼잡 정도 별로 색을 다르게 시각화하여 나타낸다. 본 연구는 1년 동안 수집한 택시 18대 데이터 및 도로 데이터를 통해 도로별 통행량을 쿼리한다. 이때 도로별 카운트 값을 기준으로 도로의 색을 구별한다. 파랑은 1개, 노랑은 2개, 빨강은 3개 이상으로 설정하였다.

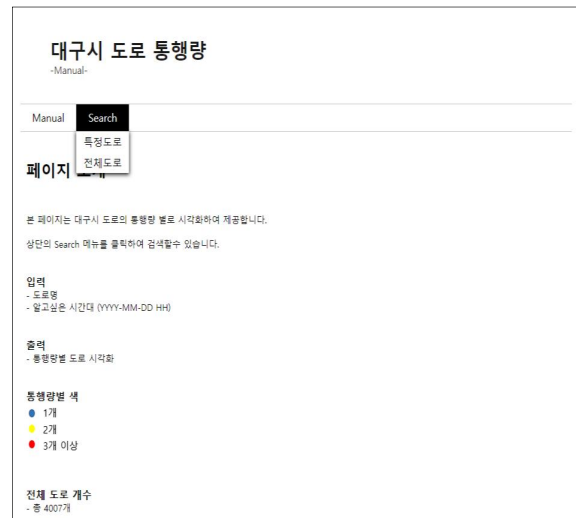


Fig. 19. Manual for Daegu Road Traffic Flow Web Page

Fig. 20은 매뉴얼 페이지에서 “Search > 특정도로” 버튼을 클릭하여 넘어온 화면이다. 지도(Map)는 도로의 통행량을 직관적으로 파악할 수 있도록 Folium에서 지원하는 오픈스트리트맵(OpenstreetMap)의 디폴트 맵(Default Map)을 사용하여 시각화하였다. 먼저 대구시 특정도로의 통행량을 쿼리하기 위해서 입력 값으로 도로명과 시작 시간, 끝 시간을 입력한다. 이때 도로명은 대구시의 도로를 선택한다. 날짜는 캘린더를 통해 입력할 수 있고, 시작 시간과 끝 시간은 00시~23시 사이로 선택할 수 있다.



Fig. 20. Traffic Flow of Specific Road Query

Fig. 21은 2017-06-03 14:00 ~ 2017-06-03 18:00의 공 원순환로에 대한 혼잡도를 쿼리한다. 이때 택시가 공 원순환로를 3번 이상 주행했기 때문에 공 원순환로에 빨간색으로 결과가 표시되었다. 이때 시각화 결과는 스크롤로 자유롭게 확대 및 축 소를 할 수 있다.

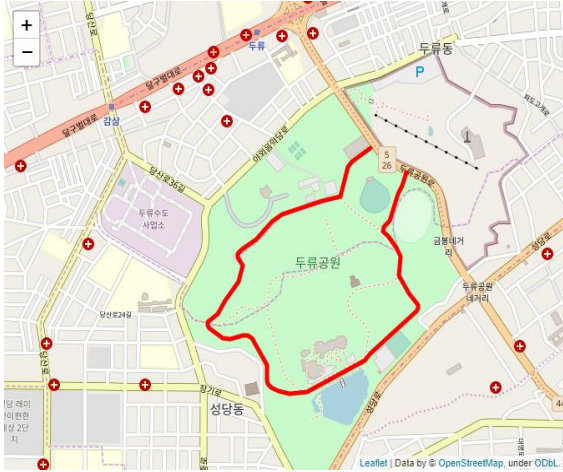


Fig. 21. Traffic Flow Result of Specific Road Query

Fig. 22는 매뉴얼 페이지에서 “Search > 전체도로” 버튼을 클릭하여 넘어온 화면이다. 먼저 대구시 전체도로 통행량을 쿼 리하기 위해 입력 값은 시작 시간, 끝 시간을 받는다.



Fig. 22. Traffic Flow of All Roads Query

예를 들어 2017-06-03 14:00 ~ 2017-06-03 18:00의 전 체 도로에 대한 통행량을 쿼리하면 Fig. 23과 같은 결과가 나

온다. 이 때 도로의 통행량이 0으로 검색되는 경우는 시각화에 표시하지 않았다. 택시는 해당 시간에 대구시 전체적으로 주행 하였고 특히 서구, 남구 근처를 주행하였다. 이때 공 원순환로, 두류공원로, 대명로, 달구벌대로, 평리로, 서대구로 등의 도로 근처는 택시가 3번 이상 주행하였기 때문에 빨간색으로 결과가 표시되었다.

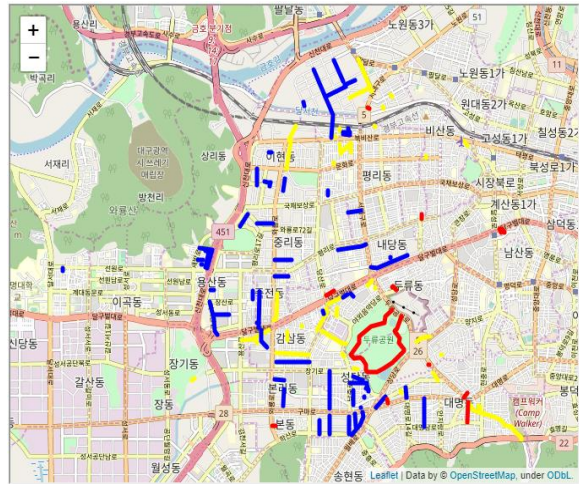


Fig. 23. Traffic Flow Result of All Roads Query

이처럼 웹 인터페이스를 통하여 원하는 시간대의 특정 도로 및 전체 도로를 쿼리할 수 있고, 검색된 시간에 해당되는 도로 별 통행량을 직관적으로 파악할 수 있다.

2.3 Road Fine Dust in Daegu

Fig. 24는 구현한 유저 인터페이스의 대구시 도로 미세먼지 웹 페이지 소개(manual)에 대한 화면이다. 매뉴얼에는 사용자 쿼리를 하기 위해 입력 값, 출력 값, 결과 값 등의 기본적인 설명을 기술하였다. “Search 메뉴” 버튼을 클릭하면 특정도로 및 전체도로 검색을 위한 페이지로 넘어간다.



Fig. 24. Manual for Daegu Road Fine Dust Web Page

도로의 미세먼지 쿼리 및 시각화는 대구시 미세먼지 농도의 분포를 한 눈에 파악할 수 있게 히트맵 분석을 목표로 시각화하였다. 먼저 대구시를 16레벨로 나눈 s2 셀들을 지도에 표시한다. 지구가 완전한 평면이 아닌 구체(sphere)이기 때문에 셀이 평행사변형으로 표현되었다.

Table 8. The Air Quality Grade by Dust Concentration

	Good	Normal	Bad	Very Bad
PM10 (mg/m^3 , day)	0~30	31~80	81~150	151~

도로가 해당하는 셀에 미세먼지 농도별 색을 구별하기 위하여 Table 8의 우리나라의 미세먼지(PM10) 기준표를 참고하였다. 파란은 0~30(좋음), 초록은 31~80(보통), 노랑은 81~150(나쁨), 빨강은 151 이상(매우나쁨)으로 설정한다.

Fig. 25은 매뉴얼 페이지에서 “Search > 특정도로” 버튼을 클릭하여 넘어온 화면이다. 지도(map)는 대구의 미세먼지 분포를 한눈에 파악할 수 있게 Folium에서 지원하는 오픈스트리트맵(OpenstreetMap)에 tiles= ‘Stamen Toner’ 옵션을 사용하여 Stamen Map으로 시각화하였다. 먼저 대구시 특정도로 미세먼지를 쿼리하기 위해서는 특정도로 쿼리의 입력 값으로 도로명과 시작 시간, 끝 시간을 입력한다. 이때 도로명은 대구시의 도로를 선택한다. 날짜는 캘린더를 통해 입력할 수 있고, 시작 시간과 끝 시간은 00시~23시 사이로 선택할 수 있다.



Fig. 25. PM10 Query of Specific Road

Fig. 26에서 2018-03-06 03:00 ~ 2018-03-06 06:00의 공원순환로에 대한 미세먼지 농도를 쿼리하면 공원순환로에 파랑색으로 표시되어 결과적으로 미세먼지 농도는 좋음(0~30)이 나왔다.

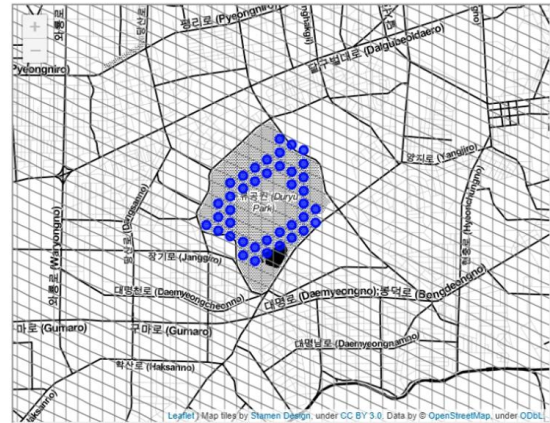


Fig. 26. PM10 Query Result of Specific Road

Fig. 27은 매뉴얼 페이지에서 “Search > 전체도로” 버튼을 클릭하여 넘어온 화면이다. 먼저 대구시 전체도로 미세먼지를 쿼리하기 위해 입력 값은 시작 시간, 끝 시간을 받는다.



Fig. 27. PM10 Query of All Roads

Fig. 28에서 2018-03-08 22:00 ~ 2018-03-08 23:00의 전체 도로에 대한 미세먼지 농도를 쿼리한 결과이다. 도로에 해당하는 셀들만 표시했기 때문에 비어있는 셀은 도로에 대한 좌표 정보가 없는 부분이다. 도로에 해당하는 셀들의 대부분은 좋음(0~30)의 파랑색과 보통(31~80)의 초록색이 나왔다. 같은 날 다른 시간대를 확인해보았을 때도 좋음(0~30)의 파랑색이 많았다. 따라서 2018년 03월 08일의 대구 미세먼지(PM10)의 농도는 대체적으로 좋음이라는 것을 알 수 있었다.

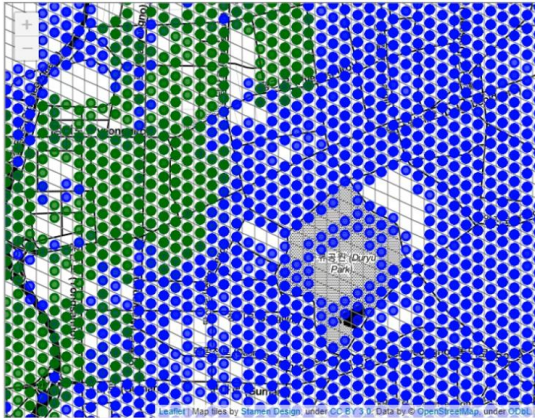


Fig. 28. PM10 Query Result of All Roads

V. Conclusions

최근 도로를 주행하는 많은 차량에 지오센서(Geosensor)들을 부착하여 다양한 사물 데이터를 수집하고 있다. 하지만 대용량 사물 데이터에 대한 진전된 검색 기술은 미흡한 실정이다. 사물 데이터의 키 값인 시간, 위도, 경도를 이용해 간단한 센서 데이터는 검색할 수 있지만 이질적인 사용자 쿼리를 처리하기 위해서는 개선이 필요하다. 또한 많은 양의 사물 데이터에서 색인을 생성하지 않고 검색하면 순차적 스캔을 통해 엄청난 시간과 비용이 낭비되어 비효율적이다. 본 논문에서는 실제 대구의 택시 18대에 센서 모듈을 설치하여 약 1년간 수집된 사물 데이터로 차량 GPS 트래젝토리를 생성하고 사용자가 도로의 혼잡 정도와 미세 먼지 농도를 빠르게 쿼리할 수 있도록 셀 기반의 공간 채움 곡선(space-filling curve) 방법을 이용해서 도로 및 차량 트래젝토리의 단일 공간 색인을 구축한다. 구축된 색인을 바탕으로 웹 유저 인터페이스에서 특정 도로와 전체 도로에 대한 트래젝토리 및 그리드 쿼리를 빠르게 요약하여 집계(aggregation)하고, 쿼리 결과는 도로별 시각화 및 히트맵의 시각적 표현을 통해 직관적으로 분석할 수 있는 웹 기반 프로토타입 시스템을 개발한다.

본 연구를 통해 시간별로 도로의 혼잡도 및 미세먼지를 파악하여 차량 정체 구간 및 미세먼지 정보 등 유용한 알림 서비스를 제공하는데 도움을 줄 수 있다. 향후에는 도로의 온도, 습도 등 다양한 쿼리 프로세싱을 지원할 수 있도록 확장할 것이다.

REFERENCES

[1] Ki Yong Lee, Minju Seo, Ryong Lee, Minwoo Park, Sang-Hwan Lee, "An Efficient Method of Processing Spatio-Temporal Joins in IoT (Internet of Things)

- Environments," Journal of KIISE, Vol. 46, No.1, pp. 88-96, Jan, 2019.
- [2] Dong-Un Lee, Yunseok Rhee, "A Multi-dimensional Query Processing Scheme for Stream Data using Range Query Indexing," Journal of the Korea Society of Computer and Information, Vol. 14, No. 2, pp. 69-77, Feb, 2009.
- [3] Eun-young Park, Young-ho Park, "A Transforming Method for Natural Environment Data Collected from IoT Sensors," Proceedings of KIIT Conference, pp. 457-460, Jun, 2018.
- [4] Ianqiu Xu, "Range Queries on Multi-Attribute Trajectories," IEEE Transactions on Knowledge and Data Engineering, VoL. 30, No. 6, pp. 1206-1211, Dec, 2018.
- [5] Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD, pp. 47-57, Jun, 1984.
- [6] Zheng Wu, "Travel Time Estimation using Spatio-Temporal Index Based on Cassandra," ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, Vol. 4, No. 4, pp. 232-242, Sep, 2018.
- [7] K. Dhanasree, C. Shobabindu, "A survey on OLAP," IEEE International Conference on Computational Intelligence and Computing Research (ICIC), May, 2016.
- [8] Joya A. Deri, Franz Franchetti, and Jose M.F. Moura, "Big Data Computation of Taxi Movement in New York City", IEEE International Conference on Big Data (Big Data), pp. 2616-2625, Dec, 2016.
- [9] Namshik Choi, Athita Onuean, Hanmin Jung, "On Visualization of Trajectory Data for Traffic Flow Simulation of Urban-scale," Proceedings of the Korean Institute of Information and Communication Sciences Conference, pp.582- 585, Oct, 2018.
- [10] Lee Minsoo, Kim Yearn Jeong, Yoon Hyejung, "Multi-query Indexing Technique for Efficient Query Processing on Stream Data in Sensor Networks," Journal of Korea Multimedia Society, Vol. 10, No. 11, pp. 1367-1383, Nov, 2007.
- [11] Road data, <http://data.nsd.go.kr/dataset/12902>
- [12] QGIS program, <https://ko.wikipedia.org/wiki/QGIS>
- [13] Hilbert curve, https://en.wikipedia.org/wiki/Hilbert_curve
- [14] Euclidian distance, <https://ko.wikipedia.org/wiki/Euclidiandistance>

Authors



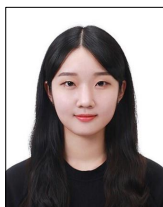
Jinju Han received the B.C. degree in Software Convergence Engineering from Kunsan National University, Korea, in 2019. Jinju Han is currently a master student in the Department of Software Convergence Engineering, Kunsan National University.

She is interested in big data system and machine learning.



Chul-Won Na is undergraduate course in Software Convergence Engineering since 2016. Chul-Won Na is currently a senior in the Department of Software Convergence Engineering, Kunsan National University. He is interested in Sentiment Crawler

and Natural Language Processing.



Dahee Lee is undergraduate course in Software Convergence Engineering since 2017. Dahee Lee is currently a student in the Department of Software Convergence Engineering, Kunsan National University. She is interested in Artificial Intelligence.



Do-Hoon Lee is undergraduate course in Software Convergence Engineering since 2018. Do-Hoon Lee is currently a student in the Department of Software Convergence Engineering, Kunsan National University. He is interested in Index

Model for Various Queries in Large-sized and Heterogeneous IoT Sensing Data.



In 2007, Byung-Won On earned his PhD degree in Department of Computer Science and Engineering at the Pennsylvania State University at University Park, PA, USA. He also received his MS degree in Department of Computer Science and Engineering at

Korea University, Seoul, Korea in 2000. Byung-Won On is an associate professor in Department of Software Convergence Engineering, Kunsan National University, Gunsan-si, Jeollabuk-do, Korea. His recent research interests are around Data Mining, Big Data, Machine Learning, and Deep Learning, mainly working on Probabilistic Topic Model, Abstractive Summarization, and Creative Computing.



Ryong Lee received the B.S. degree from the School of Electronics, Telecommunication and Computer Engineering, Korea Aerospace University, Korea, in 1998, and the M.S. and Ph.D. degrees from the Department of Social

Informatics, Kyoto University, Japan, in 2001 and 2003, respectively. From 2003 to 2008, he was a Research Staff Member at the Samsung Advanced Institute of Technology (SAIT), Korea. Since 2013, he has been with Korea Institute of Science and Technology Information (KISTI), Korea. He is currently a Senior Researcher of Research Data Sharing Center in KISTI. His research interests include spatial data analysis, the Internet of Things, smart city, and artificial intelligence.



Min-Woo Park received the B.S. and M.S. degrees from the Division of Computer Convergence, Chungnam National University, Korea, in 1992 and 2004, respectively. Since 1996, he has been with Korea Institute of Science and Technology

Information (KISTI), Korea. He is currently a Team Manager of the Research Data Sharing Center in KISTI. His research interests include system architecture, information security, the Internet of Things, smart city, and artificial intelligence.



Sang-Hwan Lee received the B.S. degree from the Department of Electronic Computing, University of Ulsan, Korea, in 1992, the M.S. degree in software engineering from Korea University, in 2004, and the Ph.D. degree from the Department

of Computer Science, University of Seoul, Korea, in 2018. Since 1995, he has been with Korea Institute of Science and Technology Information (KISTI), Korea. He is currently the director of the Research Data Sharing Center in KISTI. His research interests include big data analysis, large research data, data governance, data ecosystems, and artificial intelligence.