JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 20, No. 10, pp. 2097-2102, Oct. 2019

Inverse Haversine Method를 활용한 거리 기반 좌표 계산에 관한 연구

양 인 철 · 전 우 훈 · 문 재 필*
한국건설기술연구원 인프라안전연구본부 도로관리통합센터·차세대인프라연구센터

A study on a distance based coordinate calculation method using Inverse Haversine Method

Inchul Yang · Woo Hoon Jeon · Jaepil Moon*
Integrated Road Management Research Center & Future Infrastructure Research Center, Dept. of Infrastructure Safety Research, Korea Institute of Civil Engineering and Building Technology, Goyang-si, Gyeonggi-do, Korea

[요 약]

지구(Earth) 상의 여러 지점 간의 거리를 구하는 방식은 다양하고, 각각의 방법은 나름의 장단점을 갖는다. 그러한 방법 중 하나인 Haversine 공식은 지구를 구(sphere)로 가정하여 두 지점의 좌표(경도, 위도)를 이용하여 지구 표면을 따라 측정한 거리, 즉 대원 거리(great-circle distance)를 구하는데 활용된다. 역으로 한 지점으로부터 거리를 알고 있을 때 그 역(inverse)을 활용하여 다른 지점의 좌표를 계산할 수도 있다. 하지만 단순한 역의 공식으로는 해당 거리를 반지름으로 하는 원(circle)에 대한 공식이 유도되기 때문에 원하는 특정 지점은 구하기 어렵다. 따라서 본 연구에서는 한 지점으로부터 상하좌우로 일정 거리만큼 떨어진 지점의 정확한 좌표를 계산하는 Inverse Haversine Method (IHM) 방법을 개발하였고, 이를 상이한 좌표체계 간 좌표변환 과정에 적용하고 그 유효성을 검증하였다.

[Abstract]

A various kinds of method to calculate the distance between locations on the Earth have been studied for a long time, and each method has pros and cons. One of the proposed methods is Haversine formula which is used to calculate the grate-circle distance using the longitude and latitude of two different locations under the assumption that the Earth is sphere. Inversely, when the distance from an anchor location is known the coordinate of the other location can be calculated using the inverse of the Haversine formula. However the inverse of the formula yields only the formula of a circle with the radius of the distance, and thus it is impossible to specify the exact location. Hence, Inverse Haversine Method (IHM) was proposed to calculate the exact coordinate of a location far from an anchor location, and the performance of the method was demonstrated.

색인어 : Haversine formula, Inverse Haversine Method, 좌표변환, 교통시뮬레이션

Key word : Haversine formula, Inverse Haversine Method, Coordinate Conversion, Traffic Simulation

http://dx.doi.org/10.9728/dcs.2019.20.10.2097

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(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.

Received 16 September 2019; **Revised** 01 October 2019
Accepted 20 October 2019

***Corresponding Author; Jaepil Moon**
Tel: ※ 개인정보 표시제한
E-mail: jpmoon@kict.re.kr

Ⅰ. 서 론

우리는 하루하루의 일상에서 이동이라는 행위를 하고, 그때마다 출발지로부터 목적지까지의 거리는 중요한 관심 정보가 된다. 이뿐만 아니라 내가 사는 집에서 가까운 지하철역까지의 거리, 백화점까지의 거리, 친구 집까지의 거리와 같이 관심 지점 간의 떨어진 정도는 우리에게 중요한 정보가 아닐 수 없다. 그래서 그림 1과 같이 대부분의 웹지도에서는 거리 측정 기능을 제공하고 있다.

만약 우리가 평평한 바둑판 위에 산다면 두 지점의 좌표와 피타고라스 정리를 이용하여 쉽게 거리를 구할 수 있다. 하지만 우리가 사는 지구는 둥글다. 이동이라는 행위를 하고 있는 동안 둥근 지구의 표면 위를 지나게 되기 때문에 당연히 우리의 이동 경로는 곡선 형태가 되며, 경로의 길이를 구하기 것은 단순한 방법으로는 어렵다. 그래서 지구 상의 지점 간 거리를 구하는 다양이 방법이 연구되어 왔다[1].

거리를 구하는 방법의 입력 자료는 좌표(coordinate)이고, 좌표는 좌표체계(coordinate system)의 정의에 따라 다양한 값을 갖는다[2]. 예로부터 인류는 지구 상의 특정 지점의 위치를 정확하게 정의하고자 다양한 좌표체계 방법을 고안하였고, 각각의 좌표체계가 갖는 장단점을 활용하기 위해 변환 방법을 연구하여 왔다[3, 4].

하지만 이러한 연구는 하나의 점에 대한 변환만을 다루기 때문에 디지털 콘텐츠의 활용 측면에서 한계를 갖는다. 최근 위치 관련 디지털 콘텐츠를 다루는 많은 소프트웨어 프로그램에서는 단순하게 거리를 기반으로 하는 좌표체계를 많이 활용한다. 대표적인 예로, 교통 시뮬레이션 소프트웨어 프로그램이 있다[5]. 이는 도로 네트워크를 구축하고 그 기하구조와 운영 조건에 따라 실제 차량이 주행하는 상황을 모사(simulate)하여 도로 설계 타당성 및 교통 운영 효과, 신규 도로 타당성 등을 평가하는데 그 목적성을 갖는다. 주로 신규 주택단지 또는 대규모 상업시설 건설 시 법적으로 요구되는 교통영향평가나 신규 도로 건설, 미래 교통 체계/수단 도입 등에 따른 영향성 평가, 기존 교통체계 문제점 분석 등의 업무에 활용되고 있다.

이러한 교통 시뮬레이션 프로그램에서 주로 사용하는 좌표체계는 거리 기반의 좌표 체계이다. 이는 직교좌표계(Cartesian coordinate system)를 기반으로 x, y축으로 1 미터의 단위 길이를 이용하여 위치 정보를 구성하는 방법으로, 거리에 대한 이해가 직관적이고 현실 반영이 용이하며 복잡한 지리정보체계에 대한 이해가 불필요한 장점을 갖기 때문에 많이 활용되는 추세이다.

최근 들어 Connected vehicle 관련 연구가 활발해지며 교통 시뮬레이션 프로그램에서 현실 도로교통 상황을 모사하고, 여기서 생산된 실시간 데이터를 C-ITS (Cooperative Intelligent Transport System) 센터와 통신함으로써 그 효과를 평가하는 연구가 많이 수행되고 있다. 이 경우 거리 기반 좌표계를 갖는 시뮬레이션 프로그램과 경위도 기반의 좌표계(일반적으로 C-ITS 센터는 정밀도로지도를 이용)를 갖는 센터 간의 좌표 호환성 문제가 발생하기 때문에 교통 시뮬레이션 프로그램에서 생산되는 실시간 좌표(거리 기반 좌표)를 경위도 기반의 좌표계로 변환하는 방법이 필요하다. 이에 본 연구에서는 경위도 좌표를 이용하여 거리를 계산하는 Haversine 공식[6]을 역으로 이용하여 기준점으로부터 상하좌우로 일정 거리만큼 떨어진 지점의 정확한 좌표를 계산하는 Inverse Haversine Method를 개발한다.

다음 장에서는 Haversine Method에 대한 설명과 함께 이를 어떻게 활용하여 Inverse Haversine Method를 만들었는지에 대

(source: Kakaomap) (source: Naver map) (source: Google map)

그림 1. 각종 웹지도 서비스의 거리 계산하기 기능 예시
Fig. 1. Examples of the distance measurement of web-map services

해 기술한다. 그리고 제안된 방법의 성능을 검증하기 위해 교통 시뮬레이션 프로그램 VISSIM과 구글맵을 활용하여 테스트를 수행하고 그 결과에 대해 논의하며, 마지막으로 결론을 기술한다.

Ⅱ. Inverse Haversine Method의 개발

2-1 Haversine 공식

Haversine 공식은 지구를 구(sphere)로 가정하여 두 지점의 좌표(경도, 위도)를 이용하여 지구 표면을 따라 측정한 거리, 즉 대원 거리(great-circle distance)를 구하는데 활용된다. 대원 거리는 그림 2와 같이 구 표면의 가장 짧은 거리를 나타내며, 구 내부를 관통하는 짧은 거리와 구별하여 사용하는 개념이다.

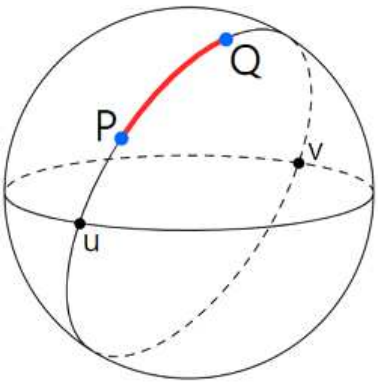


그림 2. 대원 거리 (출처: 위키피디아)
Fig. 2. Great-circle distance (source: Wikipedia)

두 점 P, Q의 경도(latitude)와 위도(longitude)를 각각 P_x, Q_x, P_y, Q_y라고 하면, 두 지점 간의 거리 d는 다음의 Haversine 공식으로 계산할 수 있다.

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{Q_x - P_x}{2}\right) + \cos(P_x)\cos(Q_x)\sin^2\left(\frac{Q_y - P_y}{2}\right)}\right) \quad \cdots\cdots (1)$$

여기서 r은 구의 반지름을 나타낸다.

2-2 Inverse Haversine Method

한 점을 중심으로 등거리의 점들을 연결하면 원이 된다. 이는 곧 중심점과 반지름으로 만들어진 고유의 원으로, 구 표면에서 중심점과의 거리가 동일한 점들의 집합을 나타낸다. Haversine 공식의 역(inverse)은 이러한 점들의 집합을 나타낸다.

하지만 본 연구에서 해결하고자 하는 문제는 특정 지점으로부터 일정 거리만큼 떨어진 어떤 지점의 좌표를 찾는 것이다. 이는 뒤에서 언급될 좌표계 변환에서 활용되는 것으로, 좌표계가 다른 계(system)의 어떤 좌표를 지구 표면의 경위도 좌표계로 변환하는 것과 동일한 문제이다.

이러한 문제의 해(solution)를 찾기 위해서는 그림 3과 같이 거리를 분할해야 한다. 구의 표면을 평면화하면 거리는 $\triangle x$와 $\triangle y$로 분할하여 표현할 수 있다. 즉 x축으로 $\triangle x$만큼 이동하고, y축으로 $\triangle y$만큼 이동하면 거리 d만큼 이동하는 것과 동일하다.

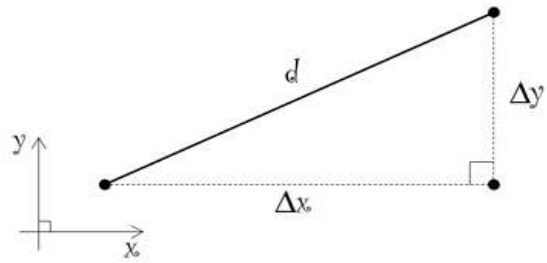


그림 3. 거리 요소 분할
Fig. 3. Element division of distance

이런 아이디어와 Haversine 공식을 이용하여 Inverse Haversine Method (IHM)를 개발하였다. IHM은 한 지점의 경위도 좌표와 x, y축 이동거리가 주어졌을 때 다른 지점의 경위도 좌표를 구하는 방법이다. 여기서 기준이 되는 지점을 Anchor point (P)라 하고 그 경위도 좌표는 P_x, P_y라 한다. x, y축 이동거리는 각각 $\triangle x$, $\triangle y$라고 한다.

식 (1)에서의 좌변, 즉 거리값이 주어지면, 우변에는 4개의 변수(P_x, Q_x, P_y, Q_y)가 존재한다. 이 중 P_x, P_y는 이미 주어진 값이기 때문에 실제 변수는 Q_x와 Q_y이다.

방정식의 해를 위한 조건은 변수와 식의 개수가 동일해야하기 때문에 각 축별 이동거리와 변수를 대응시키도록 한다. 즉, 먼저 식 (1)의 좌변에 $\triangle x$를 대입하여 우변의 Q_y를 소거하고, 그 다음으로 좌변에 $\triangle y$를 대입하여 우변의 Q_x를 소거한다. 좌변에 $\triangle x$를 대입하는 것은 y축으로의 이동이 없음을 의미하며 이는 곧 Q_y=P_y를 의미하기 때문에 Q_y가 더 이상 변수가 아닌 상수임을 의미한다. 마찬가지로 좌변에 $\triangle y$를 대입하는 것은 Q_x가 상수임을 의미한다. 따라서 두 가지 경우에 대해 각각 계산을 통해 Q_x와 Q_y를 계산할 수 있으며, 그 계산식은 다음과 같다.

$$Q_x = P_x + \frac{\triangle x}{r}\frac{180}{\pi} \quad \cdots\cdots (2)$$

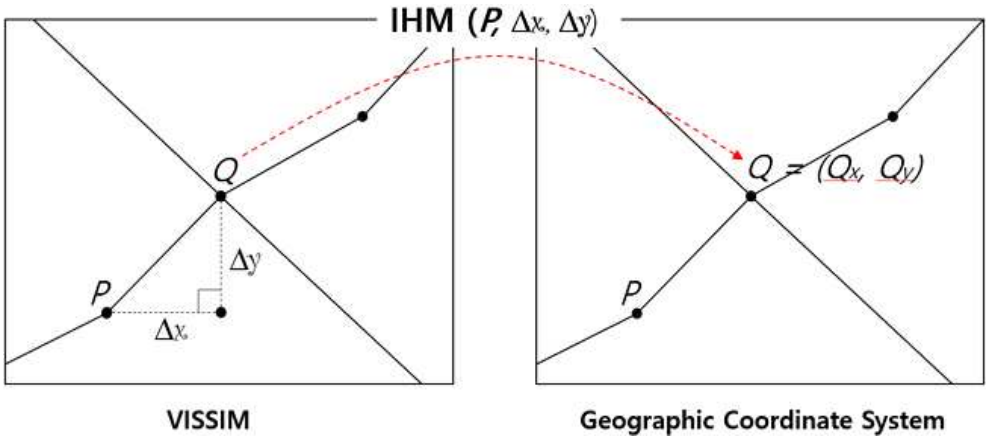


그림 4. VISSIM과 GCS 간 IHM 좌표변환 개념
Fig. 4. IHM concept applied in VISSIM to GCS coordinate conversion

$$Q_y = P_y + \frac{360}{\pi} \cdot \arcsin\left(\frac{\sin(\triangle y/2r)}{\cos(P_x)}\right) \cdots\cdots \quad (3)$$

여기서 r은 지구 반지름을 나타내고, 단위는 km이다. $\triangle x$와 $\triangle y$의 단위는 km이고, P_x, P_y, Q_x, Q_y의 단위는 degree이다.

Ⅲ. Inverse Haversine Method의 적용

3-1 개요

본 연구에서는 개발된 Inverse Haversine Method (IHM)의 성능을 검증하기 위해 이종(異種) 좌표 체계 간의 좌표 변환 문제를 활용한다.

GIS(Geographic Information System), ITS(Intelligent Transport System), LBS(Location Based Service)와 같이 공간정보를 활용하는 시스템 또는 서비스는 고유의 좌표 체계를 사용한다. 도로교통 분야에서 많이 활용되는 교통 시뮬레이션 소프트웨어 프로그램은 기본적으로 지도, 즉 공간정보를 이용하기 때문에 자체의 좌표 체계를 이용한다. 대부분의 시뮬레이션 프로그램이 거리 기반의 좌표 체계를 이용하는데, VISSIM도 마찬가지이다. 도로망을 그리기 위해 원점으로부터의 거리를 기반으로 좌표 체계를 구축한다. 즉 노드와 링크를 그리기 위해 사용되는 좌표(x, y)는 원점으로부터의 거리를 의미한다.

반면 GPS(Global Positioning System)는 경위도 좌표 체계(Geographic Coordinate System, GCS)를 이용한다. 따라서 VISSIM의 위치 정보를 GPS 데이터와 매칭하기 위해서는 거리 기반 좌표 체계를 경위도 좌표 체계로 변환해야 한다. 예를 들어 VISSIM에서 차량의 위치와 속성 정보를 추출한 후 거리 기반으로 표현된 위치 좌표를 경위도 좌표로 변환함으로써 GPS 기반의 LBS 시스템(구글맵, 카카오맵 등)과 공유가 가능하다.

따라서 본 연구에서는 그림 4와 같이 VISSIM의 차량 정보를 GCS 좌표체계를 이용하는 TRADOS 센터[7]에 전송하는 시나리오를 가정하여 IHM 방법을 이용한 좌표 체계 변환 시스템을 개발하였다. TRADOS는 레이더를 장착한 차량이 주변 차량을 인식하고 이를 센터에 전송하여 교통 상황을 추정 및 예측하는 시스템으로, 도로망의 표현 및 위치 정보 공유를 위해 경위도 좌표 체계를 이용한다. VISSIM에서 시뮬레이션 결과를 TRADOS 센터에 전송함으로써 현실에서의 구현이 어려운 시장점유율에 따른 효과 분석, 각종 도로교통 이벤트 상황에서의 성능 분석 등이 가능하다.

3-2 IHM 기반 VISSIM 좌표 변환

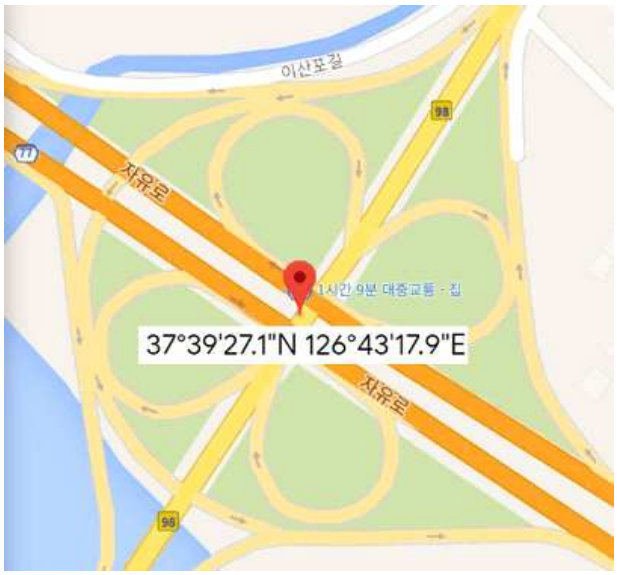


그림 5. Anchor Point 위치 (출처: 구글맵)
Fig. 5. Location of the anchor point (source: Google Map)

표 1. IHM 좌표 변환 결과
Table 1. The result of IHM coordinate conversion

No.	$\triangle x$ (km)	$\triangle y$ (km)	Q_x	Q_y	Q_x^R	Q_y^R	$RMSE_i$
1	0.5747	-0.3768	126.726807	37.654147	126.728074	37.654174	0.0009
2	1.2693	-0.8286	126.733046	37.650085	126.735824	37.650140	0.0020
3	3.2118	-1.9652	126.750496	37.639866	126.757701	37.640022	0.0051
4	3.5216	-2.1711	126.753279	37.638015	126.761309	37.638187	0.0057
5	4.6424	-2.8930	126.763347	37.631524	126.773763	37.631452	0.0074
6	5.2256	-3.2917	126.768586	37.627940	126.780473	37.628140	0.0084
7	5.4679	-3.4515	126.770763	37.626503	126.783279	37.626774	0.0089
8	6.8060	-4.4926	126.782783	37.617142	126.798570	37.617525	0.0112
9	6.9815	-4.5854	126.784360	37.616308	126.800597	37.616607	0.0115
10	7.5612	-4.6146	126.789557	37.616052	126.807030	37.616268	0.0124
11	7.9075	-5.3609	126.792656	37.609351	126.810210	37.610278	0.0124
12	8.5604	-5.6167	126.798507	37.607059	126.818298	37.607327	0.0140
13	8.2232	-6.0880	126.795468	37.602833	126.814451	37.603077	0.0134
14	9.1908	-5.9421	126.804142	37.604152	126.825218	37.604444	0.0149
15	9.5797	-6.4043	126.807619	37.600009	126.830425	37.599730	0.0161
16	10.9346	-7.9057	126.819763	37.586533	126.845094	37.586825	0.0179
17	11.8675	-8.5878	126.828118	37.580419	126.855652	37.580633	0.0195
18	12.4889	-9.0404	126.833676	37.576368	126.862653	37.576595	0.0205
19	12.6331	-9.3339	126.834951	37.573746	126.864304	37.573973	0.0208
20	13.8599	-10.0325	126.845935	37.567489	126.878057	37.567634	0.0227
21	15.0196	-10.6909	126.856314	37.561594	126.888316	37.561962	0.0226
22	15.5387	-11.2192	126.860947	37.556869	126.890420	37.557236	0.0208
23	15.2239	-11.3651	126.858103	37.555576	126.893475	37.555763	0.0250
24	14.7784	1.8670	126.854089	37.674282	126.885828	37.674035	0.0224
25	12.9945	1.3426	126.838083	37.669576	126.866586	37.669494	0.0202
26	11.3684	1.2022	126.823497	37.668315	126.848940	37.668251	0.0180
27	9.2215	0.4258	126.804249	37.661353	126.825601	37.661320	0.0151
28	8.3561	-0.3839	126.796485	37.654094	126.815880	37.653987	0.0137
29	7.9599	-1.0315	126.792925	37.648290	126.811384	37.648320	0.0131
30	7.3378	-2.3818	126.787355	37.636187	126.804355	37.636206	0.0120

IHM 방법의 적용을 위해서는 거리 기반 좌표 체계에 Anchor Point가 필요하다. Anchor Point는 좌표 체계 내에서 정확한 경위도 좌표 정보를 갖는 중심점을 의미하며, Anchor Point로부터의 거리를 활용해서 다른 지점의 경위도 좌표를 계산한다.

이를 위해 본 연구에서는 그림 5와 같이 VISSIM 도로망 내 임의의 노드, 실제 도로망에서는 자유로 이산포 IC 중심점을 Anchor Point (P)로 지정하고, 해당 지점의 경위도 좌표(P_x =37.657535, P_y=126.721644)를 입력하였다. 그리고 좌표를 얻고자 하는 지점(Q)을 선정하고, 거리 기반의 좌표 체계를 이용하는 VISSIM 도로망에서 P로부터 Q까지의 거리, $\triangle x$, $\triangle y$를 계산한 후 식(2)와 (3)을 이용하여 Q의 경위도를 계산하였다.

제안된 IHM 방법의 성능을 검증하기 위해 임의의 노드 30개를 선정하여 좌표 변환을 수행하였다. 그리고 계산된 경위도 좌표(Q)와 실제 경위도 좌표(Q^R)를 비교하여 그 성능을 검증하였는데, 에러율의 계산을 위해 RMSE (Root Mean Square Error) 인덱스를 이용하였다. 성능 검증 결과는 표 1과 같으며, 이때 지구 반지름 r은 6378.137 km를 이용하였다.

검증 결과, 대부분의 좌표 변환에서 RMSE는 0.025 이하로 나타남에 따라 변환이 잘 된 것으로 판단된다. 특히 첫 번째 좌표 변환의 경우 RMSE가 0.0009로, 거의 오차가 없는 것으로 나타났다. 하지만 $\triangle x$와 $\triangle y$가 점점 증가함에 따라 오차율이 증가하기 때문에 향후 이에 대한 보안책이 필요할 것으로 판단된다(그림 6).

Ⅳ. 결 론

본 연구에서는 위치 좌표를 아는 어떤 지점으로부터 동서 및 남북 방향으로 일정 거리만큼 떨어진 지점의 좌표를 계산하는 Inverse Haversine Method (IHM)를 개발하였다. 이를 위해 Haversine 공식에 거리 요소 분할 개념을 적용하여 역함수를 유도함으로써 대상 지점의 경도와 위도를 각각 계산하는 두 개의

식을 제안하였다. 개발된 IHM을 이용하여 거리 기반의 좌표체계를 사용하는 VISSIM 도로망의 노드 좌표를 GCS 좌표 체계로 변환하는 테스트를 수행하였고, 그 결과 IHM이 좌표 변환을 성공적으로 하는 것으로 나타났다. 다만 대상 지점이 Anchor Point와 멀어질 때 발생하는 오차율 증가는 향후 해결해야 할 문제로 판단된다.

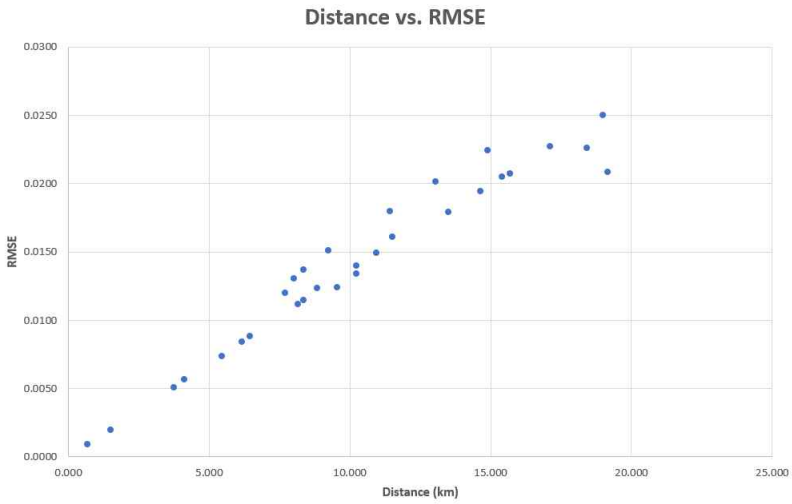


그림 6. 거리 대비 RMSE 변화
Fig. 6. Distance vs. RMSE

감사의 글

본 연구는 2019년 한국건설기술연구원 주요사업(차량센서 기반 주행환경 관측·예측·안전운행 도로기술 개발, 과제번호: 20190149-001)의 연구비 지원에 의해 수행되었습니다.

참고문헌

[1] Chamberlain, R. G., “Great circle distance between 2 point, ” http://www.movable-type.co.uk/scripts/gis-faq-5.1.html, access in Sep., 2019.

[2] Decker, B. Louis, World geodetic system 1984, Defense Mapping Agency Aerospace Center St Louis Afs Mo, 1986.

[3] Lin, K.C. and Wang, J., “Transformation from geocentric to geodetic coordinates using Newton's iteration”, *Bulletin Geodesique*, Vol. 69, No. 4, pp.300-303, 1995.

[4] Featherstone, W.E. and Claessens, S.J., “Closed-form transformation between geodetic and ellipsoidal coordinates”, *Studia geophysica et geodaetica,* Vo. 52, No. 1, pp.1. 2008.

[5] Yang, I. & Jeon, W-H, “Development of lane-close model using micro simulation”, *Journal of Digital Contents Society,* Vol. 19, No. 11, pp. 2131-2137, 2018.

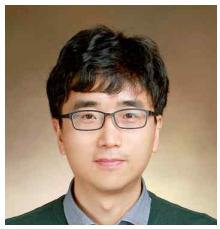
[6] Sinnott, R.W., “Virtues of the Haversine”, *Sky and Telescope,* Vol. 68, No. 2, pp.159, 1984.

[7] Yang, I. & Jeon, W-H, “Development of traffic state observation technology using automotive sensors,” *Journal of Digital Contents Society*, Vol. 19, No. 11, pp. 2131-2137, 2018.

양인철(Inchul Yang)

2011년: Ph.D. in Civil Engineering at Univ. of California, Irvine
2000년: 연세대학교 도시공학석사
1998년: 연세대학교 도시공학 학사

2011년～현 재: 한국건설기술연구원 연구위원
※관심분야 : 첨단교통, 자율주행, C-ITS, 도로안전, 도로시설



전우훈(Woo Hoon Jeon)

2016년: 서울대학교 도시계획학 박사
2001년: 한양대학교 교통공학 석사
1999년: 한양대학교 교통공학 학사

2001년～현 재: 한국건설기술연구원 수석연구원
※관심분야 : 도로안전, 무동력 교통수단, 모바일 앱, 도로시설

문재필(Jaepil Moon)

2007년: Ph.D. in Civil Engineering at NCSU
1998년: 단국대학교 교통공학 석사
1992년: 단국대학교 토목공학 학사

2009년～현 재: 한국건설기술연구원 수석연구원
※관심분야 : 도로안전, 첨단교통, 자율주행, 데이터 융합