

(KCC2015 우수논문)

대용량 GPS 궤적 데이터를 위한 효율적인 클러스터링[†]

(An Efficient Clustering Algorithm for Massive GPS Trajectory Data)

요약 도로지도 생성은 인공위성 촬영이나 현장실사를 기반으로 한다. 그리하여 도로지도를 생성하고 수정하는데 많은 시간과 비용이 든다. 이러한 이유로 차량 GPS 데이터를 이용해 도로지도를 생성하는 연구가 활발히 진행되고 있다. 도로지도 생성 연구에서 가장 중요한 문제는 주도로와 같은 대표궤적을 추출하는 것이다. 대표궤적 추출을 수행할 때에는 시작과 끝이 비슷한 궤적데이터들의 집합을 전체로 하여 궤적을 추출한다. 따라서 대표궤적을 추출하기에 앞서 전처리 과정으로 궤적 클러스터링 작업이 필요하다. 본 논문에서는 이러한 문제를 해결하기 위해 하나의 영역을 일정한 격자로 분할하였고, Sweep Line 알고리즘을 응용해 유사궤적들을 탐색한다. 마지막으로 Frechet 거리를 이용하여 궤적 간 유사도를 계산하였다. 실제로 서울의 강남구 지역에 있는 500대의 차량 GPS 궤적을 가지고 클러스터링 작업을 수행하였다. 또한, 실험을 통해 하여 격자분할 접근방식의 빠른 수행시간과 안정성을 보였다.

키워드 : 궤적 클러스터링, GPS 데이터, 도로 지도 생성, 라인 스위핑

Abstract Digital road map generation is mainly based on artificial satellite photographing or in-site manual survey work. So those map generation procedures require a lot of time and budgets to create, update road maps. For these reasons people tries to develop automated map generation systems using GPS trajectory data sets which are obtained by public vehicles. A fundamental problem in this road generation procedure is to extract representative trajectory such as main roads. Extracting a representative trajectory requires the base data set of piecewise line segments(GPS-trajectories) which have near closer starting points and ending points. So geometrically similar trajectories are selected for clustering before extracting one representative trajectory among them. This paper proposes a new divide-and-conquer approach by partitioning the whole map region into regular grid sub-spaces, then we try to find similar trajectories by sweeping. Also we applied the Frechet distance measure to compute the similarity of pair of trajectories. We conducted experiments using a set of real GPS data which has more than 500 vehicle trajectories obtained from GangNamKu, Seoul. Experiment shows that our grid partitioning approach works fast and stable to be used in real applications for vehicle trajectory clustering.

Key words : Trajectory clustering, GPS data, Road map generation, Line sweeping

1. 연구 동기

이전의 도로 생성은 항공사진과 실세계 영상분석, 직접 현장실사를 통해 이루어지며 최근 몇 년 사이에는 위성 사진을 이용한 방법이 사용되고 있다. 하지만 이러한 방법들은 초기 구축시간과 비용이 많이 드는 문제점을 지니고 있다. 거기에 더불어 새롭게 변경되는 사항들을 빠르게 반영하지 못하는 문제점이 있어 실세계와 도로가 정확히 일치하지 않을 수 있다는 문제점이 제기 되어 왔다.

하지만 최근 GPS 수신기가 탑재된 휴대단말의 보급으로 차량 내에 GPS 수신기를 장착하지 않더라도 개인 휴대단말을 이용하여 실시간으로 차량의 GPS Data 수집이 가능하게 됨에 따라, GPS 데이터를 이용하여 자동으로 도로를 생성해주는 연구가 활발히 이루어지고 있다. 이러한 도로생성 시스템에서 가장 중요하게 대두하는 문제는

여러 차량 궤적들을 중 하나의 중심 도로를 찾는 작업이다. 중심 도로를 추출하기 위해서는 시작 지점과 끝 지점이 유사한 궤적들이 필요하다. 하지만 실세계에서는 유사한 궤적들이 존재하지 않기 때문에, 중심 도로추출에 앞서 전처리 과정으로 궤적 클러스터링이 필요하다.

그림 1은 강남지역을 대상으로 택시 GPS 데이터를 받아 지도에 나타낸 그림이다. 빨간 곡선들은 실제 택시들이 지나간 GPS 데이터이며, 택시 궤적은 총 3000개이다. 이러한 GPS 데이터들을 이용하여 자동으로 도로를 찾아주면 현재 사용되고 있는 인공위성 촬영이나, 현장실사의 비용과 노력을 줄여 줄 수 있으므로, 기대하는 효과가 아주 크다.



그림 1 강남구 지역에서 수집된 택시 GPS 데이터. 빨간색 선은 총 3000대의 택시의 궤적을 나타내고 있다.
Fig. 1 Taxi GPS data collected from GangNamKu. The red line represents trajectories of 3000 taxis.

2. 관련 연구

관련 연구 [1]은 궤적 클러스터링 방법을 제시하였고, [2]는 GPS 데이터를 이용한 연구들의 성능을 비교 평가하였다. 이러한 연구 중 하나는 도로 생성 시스템으로 이 시스템은 사용자로부터 받은 GPS 데이터를 이용하여 해당 도로를 자동으로 찾아주는 시스템이다. [3]에서는 궤적들을 Clarify 하여 Routing Map을 제시하였다. [4]역시 마찬가지로 GPS 데이터를 이용해 실시간으로 도로를 생성해주는 시스템을 제안하였다. [5]에서는 Simple Median과 Homotopic Median 방법을 제시하였고, [6]에서는 Majority Median 방법을 제시하였다.

위와 같은 관련 연구는 여러 개의 궤적데이터가 유사하다고 가정하고 시작한다. 유사한 궤적 데이터란 궤적의 시작 지점과 끝 지점이 거의 비슷한 데이터들을 말한다. 아래 그림 2는 유사한 궤적에서 도로 중심선을 추출한 그림이다. 그림 3은 그림 1의 올림픽대로를 지나는 궤적 중 시작과 끝이 유사한 궤적들만 모아 지도에 나타내었다.

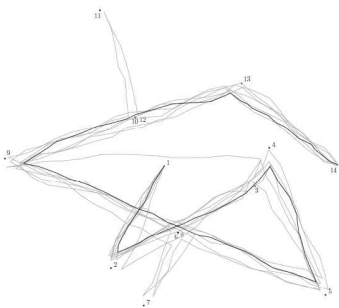


그림 2 유사한 궤적에서 도로 중심선 추출하기[6].
Fig. 2 To extract central road line from similar trajectory.



그림 3 이상적인 궤적 GPS 데이터. 검정색 선은 10개의 궤적들의 모음을 나타낸 것이며, 궤적들의 시작과 끝은 거의 유사하다.
Fig. 3 Ideal trajectory GPS data. The black line represent group of 10 trajectories, and the start to finish of trajectories is almost same.

하지만 실제 GPS 궤적데이터는 그림 4에서 보는 바와 같이 깔끔하지 않으면서 임의의 시작 지점과 임의의 끝 지점을 가지며, 유사한 궤적들만 존재할 확률은 극히 드물다. 이러한 이유로 대표궤적을 추출하기 위해서는 사전 작업으로 유사한 궤적들의 데이터 집합들을 모으는 클러스터링 기법이 필요하며, 대규모 GPS 궤적을 바탕으로 도로를 생성하는 시스템은 이러한 클러스터링 기법이 성능에 많은 영향을 끼친다. 따라서 본 연구에서는 기존연구와는 달리 실제 강남의 택시 GPS 데이터(시작 지점과 끝 지점이 유사하지 않은 데이터)를 바탕으로 대표궤적을 추출하기 위한 사전작업인 궤적 클러스터링 방법을 제안한다.

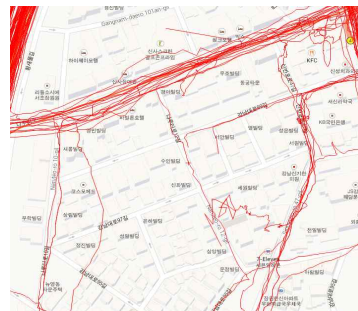


그림 4 현실 세계의 실제 궤적 데이터. 실제 GPS 차량 궤적(빨간색선)은 그림 3처럼 나타나지 않고 시작지점과 끝 지점이 다른 임의의 궤적으로 형성된다.
Fig. 4 Trajectory data of real world. Real GPS vehicle's trajectory(red line) is not showed as figure 3 but formed of any trajectory that has different start point and finish point.

3. 제안 방법

3.1 격자분할 (Grid Partitioning)

본 연구에서는 궤적 클러스터링을 수행하기 위하여 격자분할작업을 가장 먼저 수행한다. 격자분할작업이란 전체 지도의 영역을 일정한 격자 크기로 나누는 작업을 말한다. 격자분할작업을 할 때 중요한 사항은 격자의 크기이다. 격자 크기를 너무 작게 형성하여 분할을 하게 되면 한 격자 안에 세그먼트의 수가 너무 작고, 반대로 격자 크기를 너무 크게 형성하게 되면 한 격자 안에 세그먼트 수가 너무 많게 된다(여기서 말하는 세그먼트란 두 개의 GPS 좌표를 연결한 선분을 말한다). 따라서 격자의 크기는 신중히 결정하여야 한다. 본 연구에서는 격자의 크기를 신중히 결정하기 위하여 격자의 간격을 파라미터로 설정한 후, 그 간격을 1000m, 500m, 250m, 125m로 설정해 실험하여 최적의 격자 크기를 얻을 수 있었다. 그림 5는 현실 세계의 실제 궤적 데이터를 격자분할 작업을 한 그림이다.

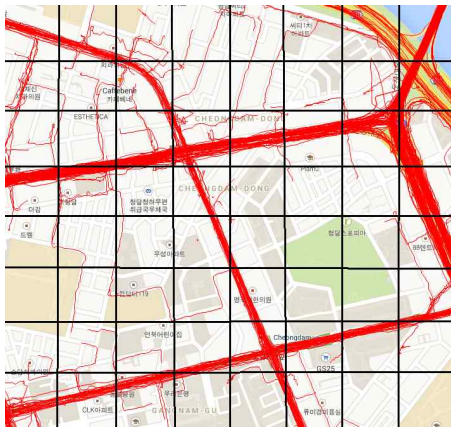


그림 5 일정한 크기로 격자 분할. 빨간색 선은 차량궤적을 나타낸 것이며, 검은색 직선은 지도를 일정한 크기로 분할 한 것이다.
Fig. 5 Grid division in regular size. The red line represent vehicle's trajectory, the black line represent baseline of map divided in regular size.

3.2 Sweep Line 알고리즘.

앞 장에서 클러스터링을 수행하기 위해 일정한 크기로 격자 분할작업을 하였다. 그 후, 각 궤적의 세그먼트가 어느 격자에 들어있는지 정보를 저장하고 있어야 한다. 우리는 이러한 정보를 Sweep Line 알고리즘을 응용하여, 빠르게 탐색을 할 수 있다[7]. Sweep Line 알고리즘을 이용하지 않고 전수조사를 하게 되면 시간복잡도는 아래와 같다.

- 세그먼트 분할 시간 : $O((H + V) \cdot S)$
- 격자 당 궤적 탐색시간 : $O(H \cdot V \cdot S)$

일반적으로 Sweep Line 알고리즘은 여러 개의 세그먼트들이 있을 때, 두 선분이 겹치는 교점의 개수를 파악할 때 많이 사용한다. 이 방식을 이용하여 먼저, 모든 세그먼트를 x축 기준으로 정렬한 뒤, 세그먼트의 시작점과 끝점의 좌표를 바탕으로 Balanced Binary Tree를 구축한다. 그 후, 구축한 Tree를 바탕으로 각 궤적의 세그먼트들이 어느 격자에 들어있는지 탐색을 할 수 있다. Balanced Binary Tree를 구축하는 시간복잡도는 아래와 같다.

- 트리 생성 시간 : $O((H + V) \cdot \log S)$
 - 격자 당 궤적 탐색시간 : $O(H \cdot V \cdot \log S)$
- {세그먼트 수 : S, 가로격자 수 : H, 세로격자 수 : V}

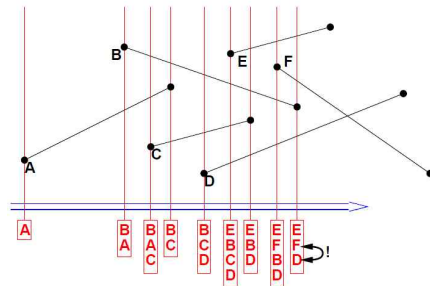


그림 6 Sweep Line 알고리즘. 여러 개의 선분이 주어질 때, 각 선분의 시작과 끝을 정렬을 하여 한 방향으로 스캔을 시도한다. 이 때 시작부분의 좌표와 끝부분의 좌표를 바탕으로 Balanced Binary Tree를 구축한다.
Fig. 6 Try scanning each segments in same direction which are lined up their start and finish, when given several segments. At this time, create Balanced Binary Tree that based on trajectories of start point and finish point.

3.3 이상 프레셰거리 평균 클러스터링

앞 장에서 클러스터링을 수행하기 위해 일정한 크기로 격자분할작업과 격자에 포함된 세그먼트의 빠른 탐색을 위해 Sweep Line 알고리즘을 이용하였다. 하지만 이러한 작업을 수행해도 한 가지 문제점이 존재한다. 아래 그림 7과 같이 하나의 격자에 여러 개의 궤적이 모여 있을 때, 유사한 궤적인지를 식별해 주어야 한다. 파란 궤적은 파란 궤적끼리 유사한 궤적으로 판별해야 하며, 빨간 궤적은 빨간 궤적끼리 유사한 궤적으로 판별하여 클러스터링을 시켜야 한다. 우리는 이러한 문제를 Average Discrete Frechet Distance(ADFD) 방법으로 해결하였다.



그림 7 그림 5의 일부 중 하나의 격자. 한 격자 안에 여러 개의 궤적들이 존재 할 때 같은 색깔의 궤적끼리 유사한 궤적이라 판별해 줘야한다.

Fig. 7 The grid, one of the figure 5. Must distinguish the same color's trajectory as similar trajectory in several trajectories from one grid.

유사한 두 임의 곡선에 대한 길이를 우리는 Frechet Distance라고 정의한다[8]. Frechet Distance의 응용인 Discrete Frechet Distance는 $O(n^2)$ 의 시간복잡도를 가진다. 하지만 [9]에서는 $O(n^2)$ 이하로 시간복잡도를 줄였다. 일반적으로 Frechet Distance를 이용할 때에는 연속된 점으로 구성되어 있지만, GPS 데이터들은 이산적으로 점이 분포되어 그 경로로 궤적을 만든다. 따라서 우리는 Frechet Distance의 응용인 Average Discrete Frechet Distance를 사용하였다. 그림 8에서 두 개의 궤적 T1과 T2가 존재한다. 두 개의 궤적의 포인트 간의 가장 짧은 거리의 합의 평균을 이용하여 Average Discrete Frechet Distance를 구한다. 본 논문에서는 각 격자마다 격자 안에 있는 모든 세그먼트들 간의 ADFD값을 비교해 그 값이 30m 이하이면 유사한 궤적이라 판별하여 클러스터링 하였다. ADFD값을 30m로 설정한 이유는 도로 폭이 약 3m인 것을 감안 했을 때 왕복 8차선을 이내에 세그먼트들이 있으면 같은 궤적이라 판단하여 30m로 설

정하였다.

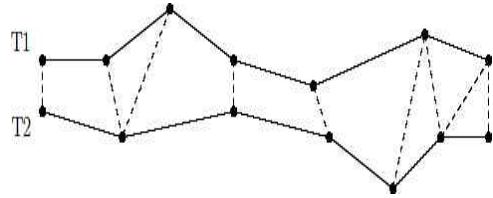


그림 8 두 궤적간의 Average Discrete Frechet Distance. 두 궤적인 T1, T2가 존재 할 때 두 궤적간의 유사도를 계산하여 일정 임계값 이하일 시 유사한 궤적이라 판단할 수 있다.

Fig. 8 Average Discrete Frechet Distance between two trajectories. We can judge two trajectories T1, T2 the similar trajectories if their similarity is under certain threshold.

4. 실험 및 평가

본 실험에 앞서 일주일간 강남의 택시 GPS 데이터를 모아 표 1, 그림 9과 같이 GPS 데이터 특징을 정리하였다. 약 16만 개의 궤적과 약 5200만 개의 포인트로 구성되어 있으며, 한 궤적 당 평균 포인트 수는 310.17이며, 한 궤적의 최대 포인트 수는 4,658개이다. 또한 한 궤적 당 최소 포인트 수는 1개이며, 포인트 수의 표준편차는 271.92이다. 마지막으로 수집한 강남구 일주일 치 GPS 데이터 파일 크기는 4.05GB로 대용량의 파일이다.

표 1 일주일간 강남구 GPS 데이터 분석.
Table 1 GPS data of GangNamKu for a week.

Trajectories	Points	Average	File size
168,236	52,183,426	310.17	4.05GB

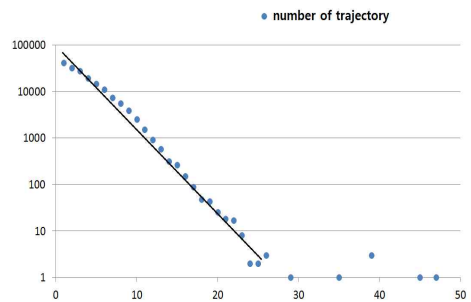


그림 9 일주일간 강남구 GPS 데이터. 가로축은 포인트 수를 나타낸 것이며, 단위 1당 100대의 차량이다. 세로축은 궤적의 수를 로그스케일로 나타낸 것이다. 포인트의 수가 100개미만인 궤적은 40,658개로 가장 많았으며, 600개 미만의 포인트 수를 가진 궤적은 전체궤적의 66.49%나 되었다.

Fig. 9 GPS data of GangNamKu for a week. The horizontal axis shows amount of point, 1 point means 100 vehicles. The vertical axis shows log scale of amount of trajectory. Trajectory that having less than 100 points made the most amounts(40,685), and trajectory that having less than 600 points took 66.49% of all trajectories.

본 실험에서는 위의 GPS 데이터 중 임의의 500개의 궤적을 바탕으로 실험하였다. 또한 GPS 데이터 영역으로 강남구 일대의 가로 6km, 세로 4km의 영역을 대상으로 하였다. 하나의 궤적은 여러 개의 포인트 세트(위도, 경도, 높이 값, 속도, 방향, 수평 정확도, 수직 정확도, 도로 네트워크와 실제 GPS 간 거리)로 구성되어 있으며, 이 포인트는 1초 간격으로 수집되었다. 강남 지역의 격자 크기를 조정하여 격자분할 작업, Sweep Line을 이용한 세그먼트탐색, 마지막으로 이산 프래셰 거리 평균 클러스터링 작업을 수행하였다.

그림 10, 11은 실험결과 중 한 격자를 나타낸 그림이다. 그림 10은 총 3개의 클러스터링 결과(주황색, 파란색, 초록색영역)를 얻을 수 있었으며, 그림 11에서는 총 2개의 클러스터링 결과(주황색, 파란색영역)를 얻을 수 있었다. 또한, 표 2와 같이 격자의 수가 증가 할수록 세그먼트 수가 증가하며, 격자 당 군집된 궤적의 수는 10x10에서 가장 많이 클러스터링 된 결과를 확인 할 수 있다. 이러한 이유는 도시 설계 계획 시 2개의 교차로 간의 길이를 약 500m로 설정하기 때문이다. 따라서 4~6km를 10x10 간격으로 나눌 시 한 격자의 간격이 약 500m가 되기 때문에 이와 부합한다고 볼 수 있다. 또한 격자의 크기를 너무 작게 하게 되면 세그먼트수가 증가해 속도가 느려지고 너무 크게 할 경우 시작과 끝이 같은 군집의 수가 적어지게 된다. 따라서 격자의 크기는 신중히 결정하여야 한다.

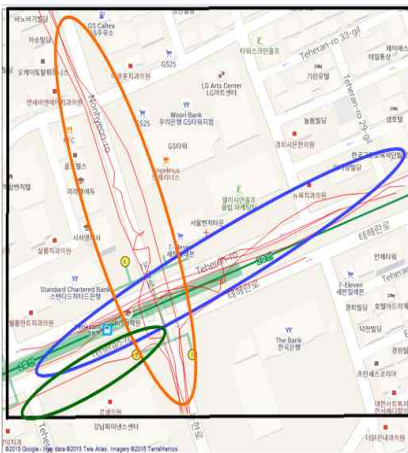


그림 10 강남구 시내를 지나는 한 격자영역을 클러스터링 한 결과이다. 세 개의 영역(파란색, 주황색, 초록색)으로 추출되었다.
Fig. 10 The result of the clustering one grid field passing GangNamKu urban. Extracted in three fields(blue, orange, green).

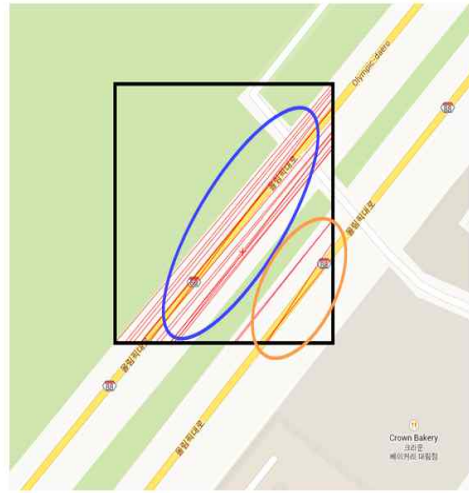


그림 11 올림픽대로를 지나는 한 격자영역을 클러스터링 한 결과이다. 두 개의 영역(파란색, 주황색)으로 추출되었다.
Fig. 11 The result of the clustering one grid field passing Olympic highway. Extracted in two fields(blue, orange).

표 2 격자수에 따른 군집된 궤적수의 변화.
Table 2 The number of trajectory depending on grid size.

Grids	Segments	Clustered trajectory	ADFD
5 x 5	17,519	2.01	30
10 x 10	18,683	3.06	30
20 x 20	20,946	1.91	30
30 x 30	23,221	1.55	30

5. 결론 및 추후 연구

본 연구에서는 일정하지 않은 궤적들의 데이터를 모아 유사한 궤적들을 클러스터링 하였다. 첫 번째 단계로는 일정한 간격으로 영역을 분할하여 효과적으로 격자마다 궤적을 분리할 수 있었고, 두 번째 단계로는 Sweep Line 알고리즘을 이용해 각 격자마다 존재하는 세그먼트들을 빠르게 탐색할 수 있었다. 마지막으로 이산 프래셰 거리를 이용해 같은 격자 안에 있는 세그먼트들을 분류해 유사한 궤적끼리 클러스터링 하였다. 이로 인해 대용량의 궤적데이터를 이용하여 도로를 생성하는 연구에서 궤적 클러스터링 작업은 꼭 필요한 작업임을 알 수 있고, 격자의 크기를 신중히 결정해야 하는 점을 알 수 있다. 또한, 실제 강남구의 택시 GPS를 바탕으로 실험한 결과 최적의 격자 크기는 10x10인 약 500m가 적절하였으며, 본 연구의 기여하는 바는 아래와 같다.

- 기존에 소개되었던 중심도로추출의 가정인 유사한 궤적(시작과 끝이 유사한 궤적)을 바탕으로 한 궤적클러스터링이 아닌 실제 강남구 전체를 지나는 택시 GPS 데이터를 이용하여 궤적 클러스터링 할 수 있으며, 이를 토대로 도로생성시스템에서 가장 중요한 중심도로를 추출하기 위한 전처리 과정을 완료하였다.

본 논문에서 제시한 방법을 이용해 클러스터링 된 궤적을 가지고 중심도로를 추출하는 것이 앞으로 해결해 나가야 할 과제이며, 강남구뿐만 아니라 더 많은 도시의 GPS 데이터를 수집하여, 가장 적절한 격자의 크기와 GPS 데이터 분석을 하는 것이 추후 연구 사항이다. 추후 중요한 문제를 정리하면 다음과 같다.

- 도로 폭(차선)에 따른 동적으로 ADFD값을 부여하는 문제
- 효율적으로 대표궤적을 추출하는 문제
- 대표궤적을 추출하였을 시 각 격자 영역의 Merge 하는 문제
- Merge 작업 후 도로형태로 보간하는 문제
- 고가도로와 터널과 같은 높이가 다른 도로 생성 적용 문제
- 생성된 도로에 대하여 방향을 부여하는 문제
- 생성된 도로의 특성을 파악하는 문제(교차로, 고가도로, 고속도로, 골목)
- 생성된 도로에 대하여 평가하는 문제

참 고 문 헌

- [1] J.-G. Lee, J. Han and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," Proc. ACM SIGMOD, pp. 593-604, 2007.
- [2] J. Biagioni and J. Eriksson, "Inferring road maps from global positioning system traces," Transportation Research Record, Vol. 2291, No. 1, pp. 61-71, 2012.
- [3] L. Cao and J. Krumm, "From GPS traces to a routable road map," Proc. ACM SIGSPATIAL, pp. 3-12, 2009.
- [4] S. Jang, T. Kim and S. Lee, "Map generation system with lightweight GPS trace data," ICACT, Vol. 2, pp. 1489-1493, 2010.
- [5] K. Buchin, M. Buchin, M. van Kreveld, M. Löffler, Rodrigo I. Silveira, C. Wenk and L. Wiratma, "Median trajectories," Algorithmica, Vol. 66, No. 3, pp. 595-614, 2013.
- [6] M van. Kreveld and L. Wiratma, "Median trajectories using well-visited regions and shortest paths," Proc. ACM GIS, pp. 241-250, 2011.
- [7] K. R. Zalik and B. Zalik "A sweep-line algorithm for spatial clustering", Advances in Engineering Software, Vol. 40, No. 6, pp. 445-451, 2009.
- [8] H. Alt and M. Godau, "Computing the Frechet distance between two polygonal curves," International Journal of Computational Geometry&Applications, Vol. 5, No. 1&2, pp. 75-91, 1995.
- [9] P. K. Agarwal, R. B. Avraham, H. Kaplan and M. Sharir, "Computing the Discrete Frechet Distance in Subquadratic Time," SIAM J. Computing, Vol 43, No. 2, pp. 429-444, 2014.

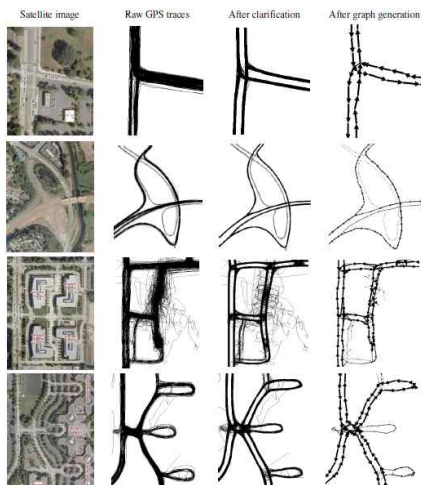


그림 12 클러스터링 된 궤적으로 중심도로 및 진행방향을 추출한 예[3].
Fig. 12 Example extracting main road and heading direction by clustered trajectory.

안녕하십니까. 먼저 2015 한국컴퓨터종합학술대회(KCC2015)에 우수 논문으로 선정해 주신 것과 정보과학회논문지의 ‘시스템 및 이론(CST)’분야에 논문을 투고할 수 있도록 해주신 것에 대해 진심으로 감사드립니다.

심사위원께서 지적해주신 여러 조언은 본 논문을 개선하는 데에 크게 도움이 되었습니다. 또한 지적해주신 내용이 모두 본 논문의 주요 문제점과 개선점에 관한 것이었음을 본 논문의 모든 저자들은 겸허히 인정합니다. 심사위원들의 지적을 바탕으로 다음과 같이 수정보안하였습니다. 수정한 내용은 아래 표와 같으며, 수정된 부분은 논문 본문에 노란색 음영(PDF 마킹 툴)사용하여 모두 표시 하였습니다.

지적하신 논문의 문제점과 요구하신 수정 내용을 반영한 수정된 결과는 아래의 표로 정리 하였습니다.

수정 요청	수정 내용
<p>지적사항1)</p> <p>1장의 서론과 관련연구를 분리하여 서술</p>	<p>1장의 서론을 연구 동기로 수정 후, 2장의 관련 연구와 분리하여 서술 -1page 왼쪽칼럼 1 ~ 오른쪽칼럼 5줄</p> <p>“이전의 도로 생성은 항공사진과 실세계 영상분석, 직접 현장실사를 통해 이루어지며 최근 몇 년 사이에는 위성사진을 이용한 방법이 사용되고 있다. 하지만 이러한 방법들은 초기 구축시간과 비용이 많이 드는 문제점을 지니고 있다. 거기에 더불어 새롭게 변경되는 사항들을 빠르게 반영하지 못하는 문제점이 있어 실세계와 도로가 정확히 일치하지 않을 수 있다는 문제점이 제기 되어 왔다. 하지만 최근 GPS 수신기가 탑재된 휴대단말의 보급으로 차량 내에 GPS 수신기를 장착하지 않더라도 개인 휴대단말을 이용하여 실시간으로 차량의 GPS Data 수집이 가능하게 됨에 따라, GPS 데이터를 이용하여 자동으로 도로를 생성해주는 연구가 활발히 이루어지고 있다.</p> <p>이러한 도로생성 시스템에서 가장 중요하게 대두하는 문제는 여러 차량 궤적들을 중 하나의 중심 도로를 찾는 작업이다. 중심 도로를 추출하기 위해서는 시작 지점과 끝 지점이 유사한 궤적들이 필요하다. 하지만 실세계에서는 유사한 궤적들이 존재하지 않기 때문에, 중심 도로추출에 앞서 전처리 과정으로 궤적 클러스터링이 필요하다.”</p>

<p>지적사항2)</p> <p>개요의 소개된 여러 논문과 제출한 논문과의 차이점이 무엇인지 알 수 없으니 1장에 기술하기 바람</p>	<p>2장 관련 연구에 기술 -2page 오른쪽칼럼 8 ~ 12줄</p> <p>“따라서 본 연구에서는 기존연구와는 달리 실제 강남의 택시 GPS데이터(시작 지점과 끝 지점이 유사하지 않은 데이터)를 바탕으로 대표궤적을 추출하기 위한 사전작업인 궤적 클러스터링 방법을 제안한다.”</p>
<p>지적사항3)</p> <p>2장에서 제안방법의 일반적인 설명들을 본 논문에서는 어떻게 적용하였는지 구체적으로 서술</p>	<p>3.1절 -3page 왼쪽칼럼 10 ~ 13줄</p> <p>“본 연구에서는 격자의 크기를 신중히 결정하기 위하여 격자의 간격을 파라미터로 설정한 후, 그 간격을 1000m, 500m, 250m, 125m로 설정해 실험하여 최적의 격자 크기를 얻을 수 있었다.”</p> <p>3.2절 -3page 왼쪽칼럼 20 ~ 오른쪽 칼럼 2줄</p> <p>“Balanced Binary Tree를 구축하는 시간복잡도는 아래와 같다.</p> <ul style="list-style-type: none"> •트리 생성 시간 : $O((H + V) \cdot \log S)$ •격자 당 궤적 탐색시간 : $O(H \cdot V \cdot \log S)$ <p>{세그먼트 수 : S, 가로격자 수 : H, 세로격자 수 : V}”</p> <p>-3page 오른쪽칼럼 10 ~ 오른쪽칼럼 14줄</p> <p>“Sweep Line 알고리즘을 이용하지 않고 전수조사를 하게 되면 시간복잡도는 아래와 같다.</p> <ul style="list-style-type: none"> •세그먼트 분할 시간 : $O((H + V) \cdot S)$ •격자 당 궤적 탐색시간 : $O(H \cdot V \cdot S)$ <p>3.3절 -4page 왼쪽칼럼 12 ~ 오른쪽칼럼 1줄</p> <p>“본 논문에서는 각 격자마다 격자 안에 있는 모든 세그먼트들 간의 ADFD값을 비교해 그 값이 30m 이하이면 유사한 궤적이라 판별하여 클러스터링 하였다. ADFD값을 30m로 설정한 이유는 도로 폭이 약 3m인 것을 감안 했을 때 왕복 8차선을 이내에 세그먼트들이 있으면 같은 궤적이라 판단하여 30m로 설정하였다.”</p>

<p>지적사항4)</p> <p>학술적으로 어떤 면에서 중요한 의미가 있는지 정리해서 기술</p>	<p>-5page 오른쪽칼럼 11 ~ 6page 왼쪽칼럼 7줄</p> <p>“또한, 실제 강남구의 택시 GPS를 바탕으로 실험한 결과 최적의 격자 크기는 10x10인 약 500m가 적절하였으며, 본 연구의 기여하는 바는 아래와 같다.</p> <p>•기존에 소개되었던 중심도로추출의 가정인 유사한 궤적(시작과 끝이 유사한 궤적)을 바탕으로 한 궤적클러스터링이 아닌 실제 강남구 전체를 지나 는 택시 GPS 데이터를 이용하여 궤적 클러스터링 할 수 있으며, 이를 토대 로 도로생성시스템에서 가장 중요한 중심도로를 추출하기 위한 전처리 과정 을 완료하였다.”</p>
<p>지적사항5)</p> <p>영문초록수정</p>	<p>첨부된 파일대로 수정</p>
<p>추가사항</p>	
<p>추후 연구사항 추가</p>	<p>-6page 왼쪽칼럼 10 ~ 왼쪽칼럼 12줄</p> <p>“강남구뿐만 아니라 더 많은 도시의 GPS 데이터를 수집하여, 가장 적절한 격자의 크기와 GPS 데이터 분석을 하는 것이 추후 연구 사항이다.”</p> <p>-6page 왼쪽칼럼 20 ~ 왼쪽칼럼 25줄</p> <ul style="list-style-type: none"> •도로 폭(차선)에 따른 동적으로 ADFD값을 부여하는 문제 •효율적으로 대표궤적을 추출하는 문제 •대표궤적을 추출하였을 시 각 격자 영역의 Merge 하는 문제 •Merge 작업 후 도로형태로 보간하는 문제 “고가도로와 터널과 같은 높이가 다른 도로 생성적용 문제 •생성된 도로에 대하여 방향을 부여하는 문제 •생성된 도로의 특성을 파악하는 문제(교차로, 고가도로, 고속도로, 골목) •생성된 도로에 대하여 평가하는 문제”
<p>추후 연구사항 그림 추가</p>	<p>그림 12</p>