

과거 궤적정보를 이용한 경로 산출 방법

Optimal Route Determination Technology based on Trajectory Querying Moving Object Database

민경욱*, 김주완, 박종현

한국전자통신연구원 텔레매틱스·USN연구단

{kwmin920, juwan, jhp}@etri.re.kr

Kyoung-Wook Min*, Ju-Wan Kim Jong-Hyun Park

Telematics & USN Research Division, Electronics and Telecommunications Research Institute

요약 위치기반서비스(LBS)에서는 사람, 차량과 같은 이동체의 위치 정보를 획득하며, 이동객체 DBMS의 개발로 이동체가 움직인 궤적정보를 효율적으로 저장 관리할 수 있다. 이동객체 DBMS에 저장된 이동체가 실제 움직인 과거 궤적정보는 다양한 형태로 가공되어 활용이 가능하다. 본 논문에서는 이동체의 과거 궤적정보를 이용하여 경로 안내를 위한 경로 산출 방법에 대하여 제안한다. 기존의 경로 안내는 교통정보 수집 장치를 통해 획득한 교통정보를 이용하여 경로를 산출하는 방식이 많이 사용되고 있어, 교통 정보가 없는 곳에서는 단순 초행길 안내 수준이다. 그러나, 본 논문에서 제안하는 방법은 고비용이 소요되는 교통정보수집장치가 없는 곳에 대해서도 과거 차량의 움직인 궤적정보를 이용하여 보다 효율적인 경로 안내를 할 수 있는 장점이 있다.

1. 서론

최근 무선 통신 및 측위 기술 등이 발전함에 따라, 위치기반서비스에 대한 관심이 높아지고 있다[1]. 위치기반서비스는 시간이 지남에 따라 위치가 변화하는 이동객체와 관련된 서비스이며, 효과적으로 서비스를 제공하기 위해서는 대용량의 이동객체에 대해서 위치획득, 저장, 질의가 가능하여야 한다. 이동객체의 과거 궤적 데이터는 다양한 형태로 사용될 수 있으며, 특히 과거에 이동객체가 이동한 데이터를 이용한 경로 안내를 위한 네비게이션 서비스에 활용할 수 있다.

네비게이션 서비스는 일반적으로 경로(route) 산출과 안내(guide)로 구성이 되며, 과거부터 경로 산출과 관련된 상당히 많은 연구들이 있어왔다. 특히, 도로 네트워크에서의 최단

경로검색 방법(shortest path problem)에 대해서 많은 연구가 진행되어 왔다[2-4]. 하지만 대부분의 경로 산출과 관련된 연구는 도로 네트워크 데이터의 정적인 속성 정보에 의존하여 최적 또는 최단 경로를 산출하게 된다. 즉, 출발점부터 목적지점까지의 최저 비용으로 이동하기 위하여, 도로 네트워크를 구성하는 도로 세그먼트(road segment)의 최고 속도, 세그먼트 길이 등의 정보를 이용하여 경로를 산출하게 된다. 목적지점까지의 예상 도착 시간은, 결과로 산출된 경로를 구성하는 각 세그먼트의 길이와 속도를 이용하여 예측할 수가 있지만, 현실 세계에서는 신뢰도와 정확도가 많이 떨어질 수 밖에 없다. 목적지점까지의 예상 도착 시간은 다양한 현실 환경 요소들에 의하여 상당히 동적으로 변화할 수 있기 때문이다.

본 논문에서는 이동객체 DBMS에 저장/관리

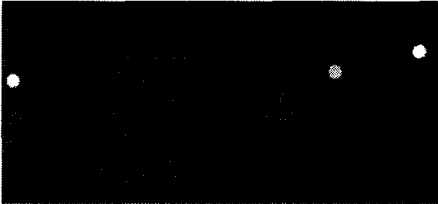
되는 이동객체의 과거 궤적정보를 이용하여 보다 신뢰성 있고, 정확한 경로를 산출하는 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 기존 경로 산출 방법의 문제점에 대해서 기술하고, 3장에서는 본 논문의 최적경로 산출을 위한 시스템 구성 및 데이터 모델에 대해서 살펴보고, 4장에서는 이동객체의 과거 궤적정보를 이용하여 최적의 경로를 산출하는 구체적인 방법에 대해서 설명하고, 마지막 5장에서는 결론 및 향후 연구과제에 대해서 논한다.

2. 기존 경로산출 방법의 문제점

최단경로 산출 알고리즘 또는 또 다른 경로 산출 알고리즘은 도로 네트워크 데이터의 정적인 속성 정보를 이용하게 된다. 예를 들어, <그림 1>에서와 같이 함수 GetRoute가 시작점 S로부터 목적지점 E까지의 경로를 산출하는 함수라고 가정했을 경우, 이 함수의 결과는 아래와 같을 수 있다.

ROUTE route =

GetRoute(S, E, RoadNetwork);



	From S To J	From J To E
Speed	60 km/hr	30 km/hr
Distance	6 km	1 km
Estimate Time	6 min	1 min

<그림 1> S→E까지의 경로 비용 산출

결과 route는 두 개의 세그먼트 S→J, J→E로 구성되고, route.Distance는 7 km, route.PassTime은 7분의 속성 값을 추출할 수 있다. 그러나, 일반적으로 커브 구간에서 이동 차량의 속도는 감속을 하게 되고, 교차로 구간에서도 차량의 속도는 감소하게 된다. 따라서, 위의 예상 경로 통과 시간 7분은 현실에서 정확한 정보일 수 없으며, 이 보다는 더 많은 시간이 소요될 것으로 예측할 수 있다. 만약, 과거에 S→E까지 지나갔

던 다수 차량의 정보를 알 수 있다면, <표 1>과 같은 형태로 구성할 수 있다.

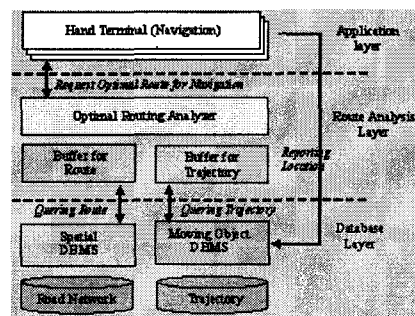
<표 1> 실제 경로[S, E]의 비용

Pass time from S to E in real world			
ID	Start Time	End Time	Pass Time (min)
A001	08:00	08:15	15 min
A003	14:13	14:25	12 min
B001	15:59	16:09	10 min
C001	18:20	18:33	13 min
C002	21:09	21:18	9 min

<표 1>과 같이 경로의 통과 시간은 시간대에 따라 다르며, 이러한 이동객체의 과거 이동 궤적정보가 충분히 많이 있다면, 좀 더 정확한 경로 정보를 제공하는데 많은 도움이 될 수가 있다.

3. 시스템 구조 및 데이터 모델

본 장에서는 이동객체 DBMS를 기반으로 최적경로를 산출하기 위한 전체 시스템 구조와 데이터 모델에 대해서 설명한다. 전체 시스템은 <그림 2>와 같이 데이터베이스 레이어, 경로 분석 레이어, 응용 레이어로 구분된다. 데이터베이스 레이어는 정적인 경로 산출을 위하여 도로 네트워크 데이터를 관리하는 공간 DBMS와 이동객체의 과거 위치정보를 관리하는 이동객체 DBMS로 구성된다. 최적경로 분석 레이어에서는 경로 산출 질의와, 과거 궤적 질의를 수행하여, 그 결과 정보인 경로와 과거 궤적정보들을 분석하여 최적경로를 산출한다. 응용 레이어는 요청된 최적경로를 네비게이션 서비스에 활용하며, 측위가 가능한 모바일 디바이스 클라이언트는 주기적으로 획득한 위치 정보를 이동객체 DBMS에 저장한다.



<그림 2> 시스템 구조

3.1 데이터 모델

최적 경로를 산출하기 위한 도로 네트워크 데이터와 이동객체 및 궤적에 대한 데이터 모델은 다음과 같다.

RoadNetwork(RN)

도로 네트워크 $RN = (S, N)$, 여기서 S는 도로 세그먼트의 집합이고, N은 연결 노드이다.

Segment

도로 세그먼트는 $(ns, ne, list\ of\ intermediate\ point, properties)$ 로 구성되며, ns 는 세그먼트를 구성하는 시작 노드이고 ne 는 마지막 노드이다. 여기서 $ns \neq ne$ 이어야 한다. ns 와 ne 를 양 끝 노드로 하는 세그먼트는 세그먼트를 구성하는 중간 포인트의 리스트로 구성이 되며, 속성 정보로 세그먼트 길이, 최고 속도, 최저 속도로 구성된다.

Node

노드는 $(p, prop)$ 로 구성되며 $p = (x, y)$, $prop$ 는 노드에 연결되어 있는 링크의 집합이다.

Moving Object

이동객체는 $(p, time, velocity, and\ segment)$ 로 구성이 된다. P 는 위치 좌표이고, $time$ 은 이동객체의 위치를 획득한 시간이고, $segment$ 는 위치를 획득한 시점에 존재하는 도로 $segment$ 정보이다.

Trajectory

궤적은 $(MBRT, list\ of\ moving\ object)$ 로 구성된다. 동일 이동객체의 과거 이동정보의 리스트와 이동객체가 이동한 전체 영역을 둘러 싸는 최소 경계 사각형과 이동한 전체 시간을 나타내는 MBRT로 구성된다. 즉, $(x1, y1, x2, y2, from, to)$ 정보로 구성 된다.

3.2 이동객체 DBMS

이동객체 DBMS는 외부 클라이언트와 SQL과 유사한 형태인 MOQL(Moving Object Query Language)을 통해 연동된다. 이동객체 DBMS는 질의처리기, 버퍼관리기, 시공간 색인, 저장관리기 등으로 구성되며, 이동객체 DBMS에 궤적 질의를 수행하여 특정 이동객체의

과거 이동 궤적을 추출할 수가 있다.

4. 과거 궤적을 이용한 최적경로 산출 방법

이번 장에서는 본 논문에서 연구한 최적경로 산출 방법에 관해서 논한다. 이에 앞서, 클라이언트에서 GPS 디바이스를 이용하여 위치를 획득하고 서버로 보고하고, 서버에서 이를 업데이트하는 프로세스에 대해서 살펴 보겠다.

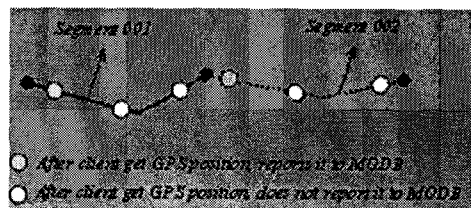
4.1 위치 보고 및 업데이트 프로세스

ClientSideProcessScenario

GPS 디바이스가 장착된 모바일 핸드셋 클라이언트는 자신의 현재 위치를 획득한 후 반드시 서버사이드의 이동객체 DBMS에 보고를 하여야 한다. 일련의 흐름은 아래의 코드와 같다.

- (1) $P_{before} \leftarrow \emptyset$
- (2) do
- (3) $P_{gps} \leftarrow GetGPS()$
- (4) $P_{current} \leftarrow CoordinateTransformation(P_{gps}, CoordType)$
- (5) $P.segment \leftarrow GetSegment(P_{current}, RN)$
- (6) if $P_{current}.segment \neq P_{before}.segment$ then Report($P_{current}$)
- (7) $P_{before} \leftarrow P_{current}$
- (8) while sleep()

모바일 클라이언트는 자신의 위치를 GPS를 통해 주기적으로 획득할 수가 있지만, 매번 획득한 위치를 서버로 보고하기에는 많은 통신 비용이 소요된다. 따라서, 본 논문에서는 <그림 3>와 같이 모바일 단말기가 획득한 새로운 위치가 이전의 위치와 동일한 도로 세그먼트에 존재한다면 보고를 하지 않고, 새로운 세그먼트에 존재할 경우에만 보고를 한다.



<그림 3> 단말에서 서버로 위치를 보고하는 경우

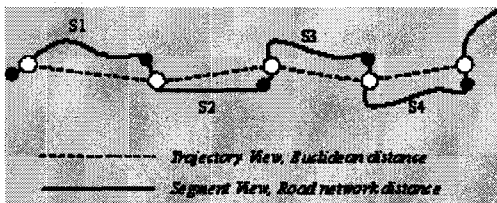
ServerSideProcessScenario

단말에서 서버로 위치를 보고한 후, 서버에서는 해당 위치정보를 다음과 같은 질의를 통해 이동객체 DBMS에 저장하게 된다.

```
INSERT INTO table (position.ID,
position.x, position.y, position.time,
position.velocity, position.segment);
```

4.2 이동궤적으로부터 도로 세그먼트 추출

이동객체 DBMS로부터 궤적 질의를 통하여 추출된 궤적정보를 가지고 실제 이동 거리를 계산할 수가 없다. 왜냐하면, 궤적정보는 비연속적인 위치정보의 리스트로 구성되어 있고, 이러한 위치정보들 사이의 거리는 실제로 도로 네트워크상의 거리가 아닌 유클리디안 거리이기 때문이다. 따라서, 과거 궤적정보로부터 실제 연속적인 이동거리를 산출하기 위해서는 각 이동 객체가 존재하였던 도로 세그먼트 정보를 추출할 수 있어야만 한다. <그림 4>는 이러한 과거궤적 거리와 실제 도로 세그먼트 거리의 차이를 보여주고 있다.



<그림 4> 과거궤적 거리와 세그먼트 거리

이동객체 DBMS로부터 추출한 과거궤적 정보로부터 실제 이동 거리와 궤적을 통과한 시간의 계산은 다음 코드로 추출할 수가 있다.

- (1) distance ← ∅ passTime ← ∅
- (2) for each mo ∈ trajectory do
- (3) segment ← GetSegment(mo.segment, RN)
- (4) distance += segment.length
- (5) passTime.from ← trajectory.MBRT.from
- (6) passTime.to ← trajectory.MBRT.to

4.3 최적경로 분석

출발지점과 도착지점을 입력 변수로 하는 기존의 경로 산출 방법들의 결과는 경로를

구성하는 기하 정보와 총 거리 및 소요 시간 정보 등으로 구성된다. 이러한 정적 경로와 비교 분석하기 위하여 이동객체 DBMS에 시작 지점과 도착지점을 지나간 과거 궤적 정보를 궤적 질의를 수행하여 추출하게 되며, 도착 지점과 시작 지점을 지나간 과거 궤적도 추출할 수 있다. <그림 5>는 궤적 질의의 예제이다.

```
Find c or which passed through from the region A to B
SELECT ID, POSITION
FROM FLEETTABLE
WHERE PASSES(POSITION, POLYGON(region A))= TRUE AND
PASSES(POSITION, POLYGON(region B))= TRUE
Find c or which passed through from the region A to B between
time t1 and t2
SELECT ID, POSITION
FROM FLEETTABLE
WHERE PASSES(POSITION, MPOLYGON(t1, t2, region A))= TRUE AND
PASSES(POSITION, MPOLYGON(t1, t2, region B))= TRUE;
```

<그림 5> 궤적 질의의 예

이러한 궤적 질의 결과 궤적 정보들은 정적 경로와 비교 분석하여 최적경로로 선택될 후보 경로들이 된다. 정적 경로와 궤적정보를 비교 분석하기 위하여, 본 논문에서는 ORDF(Optimal Route Determination Factors)를 정의한다. ORDF는 비교 분석을 하기 위한 중요한 요소이며, 보다 현실적인 최적 경로를 산출하기 위하여 다양한 분석 요소들을 정의할 수가 있다. 예를 들어, 어떤 지역에 초행인 사업가가 현재 시각이 08:30이고 A지점에서 B지점으로 이동하려고 할 경우, 자신의 차에 장착된 네비게이션을 이용하여 경로를 산출한 결과 총 거리는 10km이고 예측 소요 시간이 10분인 경로 정보가 디스플레이 될 것이다. 하지만, 이 사람이 현재 시각이 rush hour라는 사실을 몰랐다면, 그의 일정은 엉망이 될 것이다. 만약 그가 이러한 시간대에 누군가가 해당 경로를 지나갔던 경험에 대한 정보를 알고 있다면, 네비게이션 결과를 신뢰하지 않고, 다른 경로를 선택할 기회가 있었을 것이다. 즉, 과거 이동궤적에 대한 정보를 관리하는 시스템으로부터 이러한 경험정보를 추출할 수 있다면, 보다 최적의 경로를 산출하는데 많은 도움이 될 것이다. 본 논문에서 최적경로 산출에 이용한 ORDF는 <표 2>와 같다.

<표 2> ORDF

Optimal Route Determination Factor		
Factor	Description	Weight
Distance (f_0)	Road Network Distance	w_0
Pass Time (f_1)	Pass Time of consecutive segment	w_1
Time Frame (f_2)	Special time frame like rush hour	w_2
Day of week(f_3)	Special day of week like a festive day	w_3
Speed(f_4)	Keeping speed of consecutive segments	w_4
		$\sum(w_{0-4}) = 1$

<표 2>의 ORDF는 5가지 요소를 정의하고 있으며, 각 요소에는 가중치가 할당되며, 이러한 일부 요소들을 이용하여 과거 궤적 정보에 해당하는 경로비용을 산출할 수가 있다. 최적경로를 산출함에 있어, ORDF의 이용에 대해서 자세히 설명하기 위하여 다음과 같은 변수와 함수들을 정의한다.

OR : optimal Route

SR : static route

T_i : ith trajectory

NT : number of trajectory

$w(fi)$: weight of ith ORDF,

$Cost(T_i)$: route cost of ith trajectory.

식(1)과 같이 각 요소의 가중치의 합은 반드시 1이 되어야 한다.

$$f_{set} = \{f_0, \dots, f_n\}, \sum_{i=0}^n w(fi) = 1, f_{set} \subset ORDF \quad (1)$$

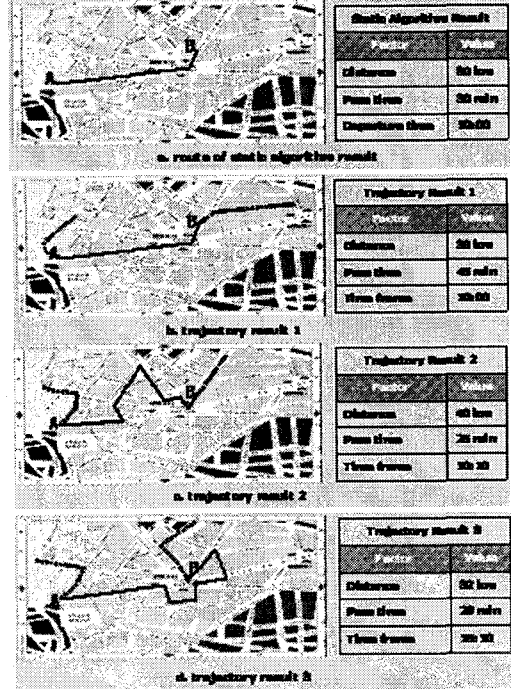
각 궤적들의 경로비용을 산출하기 위하여 궤적의 각 요소들의 값을 정규화 되어야 한다. 예를 들어 가장 간단하게, 거리 factor에 대해서는 과거 궤적들의 각 전체 거리 순서 값이 정규화 된 값으로 가정할 수 있고, time frame 요소의 경우에는 출발하려고 하는 시간 값과 각 궤적의 시작 시간과의 차이에 대한 순서 값이 정규화된 값으로 가정할 수 있다. 각 궤적에 해당하는 경로비용은 식(2)와 같이 정의할 수 있다.

$$Cost(T_k) = \sum_{i=0}^n w_i * Normalization(T_k, f_i) \quad (2)$$

마지막으로, 최적경로는 식 (3)과 같이 경로비용이 최소인 궤적이 선택된다.

$$OR = \begin{cases} \min(Cost(T_0), \dots, Cost(T_n)), & \text{if } N_T \neq 0 \\ SR & \end{cases} \quad (3)$$

<그림 6>은 정적 경로와 $ORDF = \{fd, ft, ff\}$ 를 이용하여 분석한 궤적들을 나타내고 있다.



<그림 6> 궤적 분석 결과

<그림 6>에서, f_0 는 distance, f_1 는 pass time, f_2 는 time frame으로 정의하고 있으며, 각 가중치는 $w(f_0) = 0.1$, $w(f_1) = 0.4$, $w(f_2) = 0.5$ 로 할당한다. 이 경우, f_2 time frame을 가장 중요한 요소로 간주한다. 최적 경로 분석 결과 경로 비용은 $Cost(T_1)=1.8$, $Cost(T_2)=1.7$, $Cost(T_3)=6.8$ 이며, 이 경우 각 요소들에 대한 정규화 값은 순서 값으로 정의하였다. <그림 6>에서 a의 정적 경로와 b의 궤적 1의 경우에 같은 거리 값과 유사한 time frame 값을 가지고 있지만, 경험적으로 소요시간이 45분인 것을 알 수 있다. 즉, 30분 보다는 45분이라는 정보가 현실에서 보다 더 신뢰 있는 정보이다. 일반적으로, 출발하려는 시간대와 유사한 시간대에 해당하는 과거 궤적의 거리, 통과 시간이 보다 더 신뢰가 있기 때문에 다른 요소에 비해 time frame 요소는 중요하다.

5. 결론 및 향후 연구 방향

본 논문에서는 과거 이동객체의 경험적이고 통계적인 이동 궤적 정보를 바탕으로 최적의 경로를 산출하는 방법에 대해서 기술하였다. 과거 궤적 정보는 이동객체의 과거 위치정보의 리스트로 구성이 되며, 이러한 정보는 이동객체 DBMS에 궤적 질의를 수행함으로써 추출 가능하다. 최적경로 산출을 위하여 *distance*, *pass time*, *time frame* 등의 분석 요소들을 정의하였으며, 이러한 요소들을 이용하여 과거 궤적의 경로 비용을 산출하여 최소 경로 비용에 해당하는 과거 궤적 정보가 정적인 경로 정보보다 더 정확한 정보를 제공하는 최적 경로가 될 수 있다. 따라서, 별도의 교통정보 수집장치가 없는 곳에서도 정적인 경로 계산 방법에 비해 현실에 맞는 효율적인 경로정보를 얻을 수가 있었다.

본 논문에서는 분석 요소들에 대한 정규화 값을 간단히 순서 값으로 정의를 했지만, 보다 더 정확한 최적경로를 산출하기 위해서는 체계적인 정규화 방법에 대한 연구가 필요하며, 실제 이동객체의 과거 데이터를 이용한 성능 평가에 대한 연구가 향후에는 필요하다.

<참고 문헌>

- [1] Jochen Schiller and Agnes Voisard, Location-Based Services, Elsevier Inc., 2004.
- [2] Stefano Pallottino, Maria Grazia Scutella, Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects, Technical Report, Univ. of Pisa, 1998.
- [3] Chang Wook Ahn, R. S. Ramakrishna, A Generic Algorithm for Shortest Path Routing Problem and the Sizing of Populations, IEEE Transaction on Evolutionary Computation, pp. 566-579, Vol. 6, No. 6., 2002
- [4] Stephan Winter, Weighting the Path Continuation in Route Planning, In Proceedings of 9th ACM International Symposium on Advances in Geographic information System, pp. 173-176, 2001