

이동 컴퓨팅 환경에서의 전자 수치 지도 데이터베이스를 위한 근접 최단 경로 재 계산 방법

정 성원⁰ 김 재훈
서 강 대학교 컴퓨터 학과
jungsung@ccs.sogang.ac.kr freeso@mclab.sogang.ac.kr

An Approximate Shortest Path Re-Computation Method for Digital Road Map Databases in Mobile Computing Environments

Sungwon Jung⁰ Jaehun Kim
Dept. of Computer Science, Sogang University

요 약

이동 컴퓨팅(Mobile Computing)의 상업적인 응용분야로서, 지능형 교통정보시스템(ITS)에서의 첨단 여행자 정보시스템(ATIS)이 있다. ATIS에서 가장 중요한 이동 컴퓨팅 태스크는 현재 위치에서 목적지까지의 최단 경로를 계산하는 일이다. 본 논문에서는 최단 경로 재 계산 문제에 대해서 연구하였다. 이 문제는 전자 수치 지도(topological digital road map)상의 간선(edge) 비용이 동적인 교통 상태에 따라 빈번하게 갱신되고 있는 ATIS의 동적 경로 안내 시스템(DRGS)에서 발생한다. 지금까지 제안된 방법들은 처음부터 최단 경로를 재계산하거나, 또는 단지 비용의 변화가 일어난 간선 상에 있는 양 끝 노드 사이에 대해서만 최단 경로를 재계산할 뿐이다. 본 논문에서는 앞서 계산된 최단 경로에 대한 정보를 이용하는 효율적인 적용형 슬라이딩 윈도우 기반의 근접 최단 경로 재 계산 방법을 제안한다.

1. 서 론

무선통신 기술의 발달과 함께, 고성능 휴대용 컴퓨터의 등장은 이동 컴퓨팅의 구현을 현실화시키고 있다. 이러한 이동 컴퓨팅의 상업적인 응용분야로서 지능형 교통정보시스템(ITS)의 첨단 여행자 정보시스템(ATIS)이 있다. ATIS에서 가장 중요한 이동 컴퓨팅 태스크는 현재 위치에서 목적지까지의 최단 경로를 계산하는 일이며, 여기서 중요한 문제는 ATIS가 실시간 이동 시스템이기 때문에 어떤 경로를 제한된 시간 안에 계산해야 한다는 사실이다.

전자 수치 지도 데이터베이스 분야에서 최단 경로 계산 문제에 중점을 둔 두 가지 형태의 연구 결과가 문헌을 통해 알려져 있다. 그중 하나는 대략적인 최단 경로를 신속히 제공하는 데이터베이스 체계를 개발한 것이고[1], 다른 하나는 최단 경로 계산을 빠르게 할 수 있는 전자 수치 지도를 구성하기 위한 효율적인 데이터베이스 구성 방법을 개발한 것이다[2].

본 논문에서는 최단 경로 재 계산 문제에 대해서 연구하였다. 이 문제는 전자 수치 지도상의 간선 비용이 동적인 교통 상태에 따라 빈번하게 갱신되고 있는 ATIS의 동적 경로 안내 시스템(DRGS)에서 발생한다. 교통 상태가 동적으로 변화하기 때문에, 효율적인 최단 경로 재 계산 방법은 DRGS의 성공적인 수행을 위해 필수적이다. 현재 사용되고 있는 두 가지 방법은 앞서 계산된 최단 경로 정보를 재사용하지 않는다는 점에서 비효율적이다. 계산된 경로 정보를 이용하여 앞서 계산된 최단 경로를 부분적으로 바꾸는 방법은 새로운 현재 위치에서 완전한 경로를 계산하는 것보다 더 빠를 것이다.

본 논문에서는 앞서 계산된 최단 경로의 정보를 이용하는 효율적인 적용형 슬라이딩 윈도우 기반의 근접 최단 경로 재 계산 방법을 제안한다. 이 방법은 지도상의 임의의 두 점점사이 에 대한 최적의 최단 경로를 보증하지는 않지만, 최적에 상당히 가까운 경로를 제공한다. 제안한 방법의 기본적인 가정은 실제 교통상황에서 경로 재 계산에 드는 매우 큰 시간 비용을 감안할 때, 운전자들은 최적으로 재 계산된 최단 경로보다 신속히 계산되는 대략적인 최단 경로를 얻고자 한다는 것이다.

2. 최단 경로상의 간선 갱신에 대한 영향 분석

이 장에서는 최단 경로상의 하나의 간선에 비용 갱신이 일어났을 경우 이 갱신이 최단 경로에 어떤 영향을 미치는지 분석한다. 먼저, 최단 경로를 계산하기 위해선 A*와 OTO 알고리즘을 고려해 볼 수 있다. 그림 1은 출발지(s)와 목적지(d)에 대한 두 알고리즘의 탐색 공간 분포와 탐색 방향을 나타내고 있으며, 탐색 공간이 더 작은 OTO가 A*보다 효율적이라는 사실을 알 수 있다. 따라서 본 논문에서는 최단 경로를 계산하는데 있어 OTO를 채택한다. 그리고, 두 알고리즘의 pseudo코드에서는 어느 시점까지 조사된 탐색 공간의 크기를 frontierSet이라는 데이터로 유지하고 있으며[3], 이는 2.2절에서 다시 언급한다.

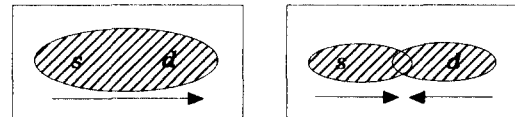


그림 1 A*와 OTO에 대한 조사된 탐색 공간의 상대적인 크기

다음으로, 최단 경로의 계산은 4개의 인접 노드를 가진 2차원 격자 그래프 $G(V,E,L)$ 상에서 수행된다. 여기서, V 는 노드 집합, E 는 간선 집합, L 은 비용의 집합이며, 각각의 간선 (u,v) 는 비용 $L(u,v)$ 와 결합된다.

2.1 최단 경로 재 계산에서 경로 재 사용률 측정

이 절에서는 앞서 계산된 최단 경로와 장애가 발생한 후 처음부터 다시 재 계산된 최단 경로와의 중복도를 실험한다. 먼저, 이 절에서 사용될 다음 두 정의를 살펴본다.

정의 1. $SP(x,y)$ 가 $G(V,E,L)$ 에서 출발지 x 로부터 목적지 y 까지의 최단 경로라고 하자. 이때, $SP(x,y)$ 는 $z_0=x, z_m=y$ 인 노드 수열 $(z_0, z_1, \dots, z_{m-1}, z_m)$ 와 같이 순차적으로 표현할 수

있다. $0 \leq i \leq m-1$ 일 때, $SP(x, y)$ 상의 어떤 간선 (z_i, z_{i+1}) 의 비용이 갱신된 후, 출발지 x 에서 목적지 y 까지의 새로 재 계산된 최단 경로가 $N_SP(x, y)$ 라고 하자. 마찬가지로, $N_SP(x, y)$ 는 노드 수열 $(w_0, w_1, \dots, w_{n-1}, w_n)$ 와 같이 표현할 수 있다. □

정의 2. $SP(x, y) \neq N_SP(x, y)$ 인 $SP(x, y)$ 와 $N_SP(x, y)$ 가 주어졌다고 하자. i 를 0에서 $n-1$ 까지 증가시키면서, $z_{i+1} \neq w_{i+1}$ 일 때까지 조건 $z_i = w_i$ 을 만족하는 노드 개수를 L 이라고 하자. 또, $i=n$ 에서 1까지 감소시키면서, $z_{i-1} \neq w_{i-1}$ 일 때까지 조건 $z_i = w_i$ 을 만족하는 노드 개수를 R 이라고 하자. 이때, 경로 재 사용률 $PRP(x, y)$ 를 $(L + R) \times 100 / m$ 으로 정의한다. □

본 논문에서는 실험을 위해 $|V|=50 \times 50$ 개의 노드, $|E|=4 \times 50 \times 49$ 개의 방향 간선과 각각 20%, 40%, 80%, 160%, 240%, 320%의 간선 비용의 다양성(variations of edge costs)에 대응하는 6개의 균등한 분포 [100,120], [100,140], [100,180], [100,260], [100,340], [100,420]에 기반하여 간선 비용을 생성한 총 6×100 개의 격자 그래프 $G(V, E, L)$ 를 생성하였다(각 간선 비용의 다양성에 대해 100개 씩 생성). 그 다음, 출발지와 목적지로서 5개의 노드 쌍 즉, 그림 2와 같이 $\theta=15, 30, 45$ 도를 이루며 마주보는 3개의 노드 쌍, 수직으로 마주보는 1개의 노드 쌍($\theta=90$), 수평으로 마주보는 1개의 노드 쌍($\theta=0$)을 선택하였다.

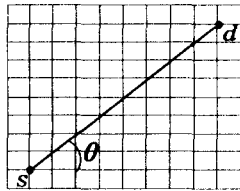


그림 2 출발지(s)와 목적지(d)가 이루는 각도 θ

각 격자 그래프에 대해서, 5개의 노드 쌍에 대한 5개의 최단 경로를 계산하고, 각 최단 경로의 5%, 10%, ..., 90%, 95%에 해당되는 위치의 간선 비용을 갱신하였다. 그리고, 각 해당 위치의 간선 비용 갱신에 대하여 $PRP(x, y)$ 를 계산하였다. 계산된 $PRP(x, y)$ 는 각 간선 비용의 다양성에 따라 100개의 격자 그래프에 대해서 평균화한다. 실험결과를 그림 3에서 보인다.

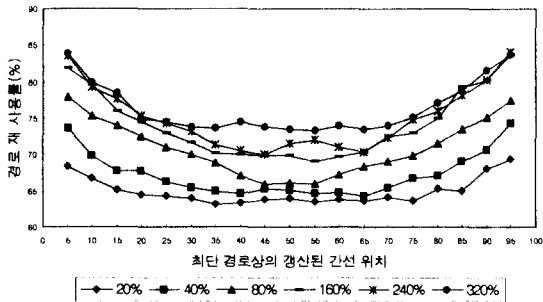


그림 3 최단 경로상의 각 위치에 대한 $PRP(x, y)$

여기서, 다음의 세 가지 흥미로운 특성을 발견할 수 있다:

1. $PRP(x, y)$ 는 간선 비용 분포에 따라 60%~90%까지 변한다.

2. 간선 비용 분포가 20%에서 320%까지 증가함에 따라, $PRP(x, y)$ 도 역시 증가한다.
3. 각 간선 비용 분포에 대해서, 그것의 $PRP(x, y)$ 는 최단 경로의 양 끝 노드에서 최대이고 중간 위치의 노드로 갈수록 감소하기 시작하여, 중간 노드에서 최소가 된다.

특성 1로부터, 앞서 계산된 최단 경로상의 대부분의 노드들이 간선 비용이 갱신된 후 새롭게 재 계산된 최단 경로에서 재 사용된다고 결론 내릴 수 있다. 특성 2와 특성 3에 대해서는 경로 재 사용률 $PRP(x, y)$ 에 영향을 미치는 벡터들을 결정하기 위한 보다 자세한 분석이 필요한데, 이를 각각 2.2절에서 논의한다. 만약, 이 벡터들에 대한 정보를 이용한다면 제안한 방법의 목표인 근접 최단 경로를 효율적으로 산출할 수 있다.

2.2 경로 재 사용률에 영향을 미치는 벡터 분석

먼저, 특성 3을 설명하기 위해서 앞에서 언급한 $frontierSet$ 을 살펴본다. 2.1절과 같은 격자 그래프 $G(V, E, L)$ 상에서 $\theta=45$ 인 노드 쌍(예를 들어, (10,10), (31,31))에 대한 최단 경로를 계산하면서 $frontierSet$ 의 크기를 측정할 수 있다. 이 실험에서, G 의 간선 비용 분포는 80%로 고정한다. 결과를 그림 4에서 보이며, 예상했던 대로 $frontierSet$ 의 크기는 최단 경로상의 양 끝 노드에서 최소이고 중간 노드로 갈수록 증가하여, 중간 노드에서 최대가 된다.

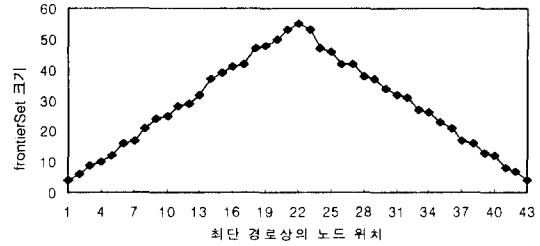


그림 4 최단 경로상의 각 노드에 대한 $frontierSet$ 크기 변화

위 사실은 양 끝 노드에서부터 중간 노드에 가까울수록, 양 끝 노드에 비해서 선택할 수 있는 최단 경로(대안 경로)가 많아진다는 사실을 암시한다. 대안 경로의 증가는 최단 경로와 대안 경로간의 중복도가 낮아진다는 것을 의미하며, 결과적으로 $PRP(x, y)$ 의 값은 낮아진다.

다음 특성 2를 설명하기 위해, 2.1절과 같은 간선 비용의 다양성이 20%, 80%, 320%인 3×100 개의 격자 그래프 상에서 두 개의 노드 쌍(수평, 45도)을 선택한 후, 두 노드 쌍 각각에 대해서 2.1절과 같은 방법으로 $PRP(x, y)$ 를 측정하였다. 그 결과를 그림 5에서 보인다.

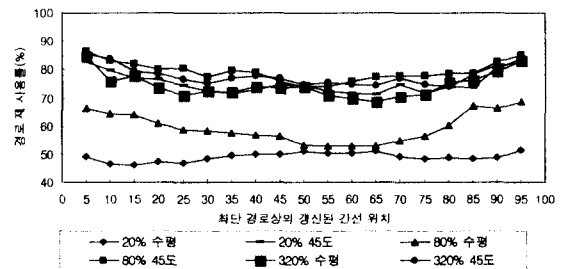


그림 5 수평, 45도 경로에 대한 $PRP(x, y)$

위의 결과 그래프에서, 간선 비용의 다양성에 따른 $PRP(x, y)$ 의 변화는 수평, 수직 노드 쌍(수직과 수평은 격자 그래프 상에서 기본적으로 동일하다.)에 대한 $PRP(x, y)$ 의 차이에서 오는 것을 알 수 있다. 즉, 수평, 수직 노드 쌍에 대한 $PRP(x, y)$ 는 간선 비용의 다양성이 증가할수록 높아진다. 따라서, 출발지와 목적지가 이루는 각도와 간선 비용의 다양성이 경로 재 사용률 $PRP(x, y)$ 에 영향을 준다는 사실을 알 수 있다.

3. 적응형 슬라이딩 윈도우 기반의 근접 최단 경로 재 계산 방법

본 논문에서는 2장에서 언급한 세 가지 특성을 이용하는 동적 윈도우 기반의 근접 최단 경로 재 계산 방법을 제안한다.

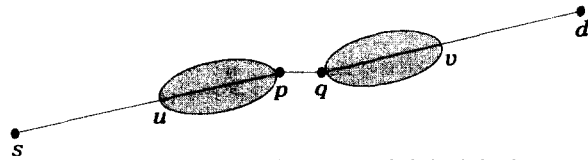


그림 6 출발지 s에서 목적지 d까지의 최단 경로

그림 6은 제안한 방법의 기본 개념을 보여준다. 여기서, 출발지 s에서 목적지 d까지의 최단 경로 상에 간선 (p,q)의 비용이 갱신되었다고 가정한다. 2장에서 특성 1로부터, 최단 경로상의 어떤 노드 s와 v에 대해서, s에서 u까지 그리고 d에서 v까지의 두 간선(예를 들어, 노드 수열)이 s에서 d까지 재 계산된 최단 경로에서 재 사용될 것이다. 만약 노드 u와 v를 정확히 찾을 수 있다면, 다음 두 과정과 같이 최단 경로를 재 계산할 수 있다:

1. 노드 u에서 v까지의 최단 경로 $SP(u, v)$ 만을 A^* 또는 OTO 알고리즘을 이용하여 계산한다.
2. $N_SP(s, d)$ 는 세 경로, $SP(s, u)$, $SP(u, v)$, $SP(v, d)$ 들을 연결한 것이다. 즉, $N_SP(s, d) = SP(s, u) \parallel SP(u, v) \parallel SP(v, d)$

위의 방법으로 앞서 계산된 최단 경로 상에 장애가 발생했을 때, 출발지에서 목적지까지의 최단 경로를 처음부터 다시 계산할 필요 없이 노드 u와 노드 v사이의 최단 경로만을 구하여 앞서 계산된 최단 경로의 두 부분(예를 들어, $SP(s, u)$ 와 $SP(v, d)$)과 연결만 하면 된다. 이는 최단 경로를 구하는데 있어 부담해야 하는 많은 계산 비용을 줄일 수 있다.

제안한 방법에서 사실상 가장 중요한 작업은 실제 최단 경로에 최대로 근접하기 위한 적당한 윈도우 크기(최단 경로를 새롭게 계산해야 하는 구간 즉, 노드 u에서 노드 v까지의 구간)를 동적으로 결정하는데 있다. 먼저 제안한 방법은 윈도우를 찾는데 있어, 최단 경로상의 각 노드들에 대한 $frontierSet$ 크기를 이용한다. $|frontierSet_p|$ 을 노드 p에서의 $frontierSet$ 크기라고 하자. 이때, $G(V, E, C)$ 상에 그림 7과 같이 타원 내부의 총 노드 개수가 $|frontierSet_p|$ 에 가장 근접하도록 노드 p와 u가 초점인 타원을 하나 그림으로써 노드 u를 결정한다.

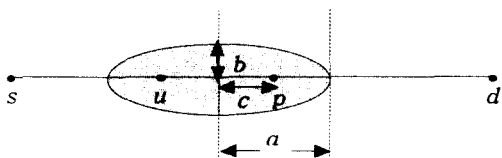


그림 7 격자 그래프상에 생성된 타원의 형태

여기서, 타원의 크기를 결정하는 비율 a/b 의 적절한 선택이 중요한데, $frontierSet$ 크기와 함께 $PRP(x, y)$ 에 영향을 주는 두 벡터 즉, 출발지와 목적지가 이루는 각도와 간선비용의 다양성을 고려하여 결정하여야 한다. 예를 들어, 출발지와 목적지 사이의 각도가 $\theta=0$ 이고 간선 비용의 다양성이 20%이라면, 윈도우를 크게 결정하기 위해 비율 a/b 를 크게 해야한다. 비율 a/b 의 효율적인 결정 방법은 아직 연구 중이며, 제안한 방법의 효율성을 확인하기 위해 일단 임의의 값을 주어 실험하였다. 다음에서 노드 u를 찾는 보다 자세한 알고리즘을 보인다:

1. $SP(p, s)$ 를 $z_0=p$, $z_m=s$ 인 노드 수열 $(z_0, z_1, \dots, z_{m-1}, z_m)$ 으로 표현하자. 또, $i=0$ 이라고 하자.
2. $i=i+1$
3. $E(p, z_i)$ 를 p와 z_i 사이의 유클리드 거리라고 하자.
4. 타원 $x^2/a^2 + y^2/b^2 = 1$ 을 생성한다. 여기서, $c^2 = a^2 - b^2$ 이고 타원의 초점이 $F = z_i$, $F' = p$ 이므로, $c = E(p, z_i)/2$ 이다. 이 알고리즘에서 비율 a/b 는 1.1, 1.2, 1.3, ..., 2.0 중 하나로 결정한다.
5. 타원 내부의 총 노드의 개수를 N 이라고 하자.
만약 $N = |frontierSet_p|$ 이면, 적당한 노드 $u = z_i$ 를 찾은 것이고 종료한다. 만약 $N < |frontierSet_p|$ 이면, $N' = N$ 이고 2 단계로 간다.
만약 $N > |frontierSet_p|$ 이면, 다음과 같이 노드 u를 결정한다.
 $|N - |frontierSet_p|| > |N' - |frontierSet_p||$ 이면 $u = z_{i-1}$ 이고, 그렇지 않으면 $u = z_i$ 이다.

그림 8 노드 u를 찾는 알고리즘

4단계와 5단계에서 사용된 휴리스틱은 노드 u에서부터 p까지 생성된 탐색 공간의 크기가 $|frontierSet_p|$ 에 근접하도록 최단 경로상의 노드 u까지 후퇴할 필요가 있다는 것을 의미한다. 탐색 공간이 타원의 형태로 확장되기 때문에, 여기서 타원 방정식을 사용한다. 이와 유사한 방법으로 노드 v도 결정할 수 있다. 노드 u와 노드 v에 대한 보다 자세한 결정 방법과 제안한 방법의 효율성 실험 결과는 [3]을 참고하기 바란다.

4. 결론

본 논문에서는 앞서 계산된 최단 경로 정보를 이용하는 효율적인 적응형 슬라이딩 윈도우 기반의 근접 최단 경로 재 계산 방법을 제안하였다. 제안한 방법은 면밀한 실험을 통한 최단 경로상의 갱신 영향의 분석에 그 근거를 두고 있다. 이 방법으로 인해 처음부터 다시 최단 경로를 재 계산하는데 드는 많은 시간 비용을 절감할 수 있다. 현재 제안한 방법의 효율성을 높이기 위해서, 타원의 크기를 좌우하는 비율 a/b 의 결정 문제와 여러 간선 비용의 갱신을 다루기 위한 연구가 진행 중에 있다.

5. 참고 문헌

- [1] Y. Huang, N. Jing, and E. Rundensteiner, Hierarchical Path Views: A Model Based on Fragmentation and Transportation Road Types, In Proc. of the 3rd ACM Workshop on Geographic Information Systems, pp. 93-100, 1995.
- [2] R. Goldman, N. Shivakumar, S. Venkatasubramanian, and H. Gracia-Molina, Proximity Search in Databases, In Proceedings of the 24th VLDB Conference, pp. 26-37, 1998.
- [3] S. Jung and J. Kim, An Approximate Shortest Path Re-Computation Method for Digital Road Map Database in Mobile Computing Environments, Technical Report, 2001