유비쿼터스 환경에서의 이동 객체 위치 추적 방법 비교를 위한 테스트 베드 시스템

한득춘, 김시완, 이기준
부산대학교 컴퓨터 공학과
{dchan, swkim}@isel.cs.pusan.ac.kr lik@pnu.edu

In Ubiquitous Environment, Test Bed System for Comparison of Moving Objects Position Tracking Methods

Deck-Chun Han, Si-Wan Kim, Ki-Joune Li
Department of Computer Engineering, Pusan National University

요 약

유비쿼터스 환경에서 모든 이동 객체들의 정확한 위치를 추적하는 것은 현실상 불가능하므로, 현실적인 대안으로 위치 추적 방법을 사용 한다. 현재 위치 추적 방법들은 많이 개발 되고 있지만 이것을 비교 실험할 수 있는 환경이 미흡한 실정이다. 이에, 본 논문에서는 여러 가지 위치 추적 방법을 비교 실험 할 수 있는 테스트 베드 시스템을 구현하였다. 또한 본 논문에서 구현한 테스트 베드 시스템에서 현재 나와 있는 여러 가지 위치 추적 방법을 실험을 통해 비교, 분석해 보았다.

1. 서론

이동 객체들의 위치 추적은 위치 기반 서비스(LBS)에서 중요한 부분이다. 가장 이상적인 방법은 이동 객체들이 현재 위치를 서버에 실시간으로 알려주고, 서버는 그 정보를 받아 위치를 알아내는 것이지만, 이동 객체들의 개수 많아지면 서버에서 실시간으로 모든 정보를 처리하기 힘들다[6]. 따라서 갱신 간격을 늘림으로써 해결할 수 있고, 갱신 간격을 결정하는 기준과 갱신 간격 사이에 존재하는 위치 정보의 추정 방법이 위치 기반 서비스 품질을 결정하는 중요한 요소이다.

현재 여러 가지 위치 추적 방법들이 개별적으로 연구되고 있으나, 이러한 방법들을 비교 할 수 있는 환경이 미흡한 실정이다. 이에 본 논문에서는 여러 가지 위치 추적 방법을 비교 실험 할 수 있는 테스트 베드 시스템을 구현하였다. 본 논문에서는 기존에 제시된 3가지 위치 추적 방법[1, 5]을 소개하고 테스트 베드 시스템을 이용하여 각 위치 추적 방법들의 성능 비교 분석하였다. 특히 도로 정보가 있는 위치 추적 방법을 사용할 경우, 도로를 따라 가장 가까운 것을 찾는 질의를[3] 처리할 수 있다.

2. 위치 추적 방법들

2.1 점 기반 위치 추적 방법

점 기반 위치 추적 방법은 가장 간단하고 이해하기 쉬운 방법으로 현재 위치 정보만 가지고 추적 가능하다.

이 방법의 알고리즘을 살펴 보면, 최초 이동 객체가 서버에 등록할 때, 위치를 자신도 저장하고 서버에도 위치를 등록한다. 이동 객체는 실시간으로 GPS(Global Positioning System) 신호를 받아서 저장된 위치와 현재 위치를 비교한다. 현재 위치와 저장된 위치 사이의 거리가 오차 임계 값(threshold)을 넘었을 경우, 이동 객체는 서버에게 현재 위치를 보내어 자신의 위치를 갱신한다. 이 방법의 갱신은 임계 값의 거리 마다 발생하고, 서버에 저장된 위치가 현재 이동 객체의 위치라고 추정한다.

이 방법은 속도가 빠르지 않은 객체나 이동이 많지 않은 객체에 적당하다. 이동 객체가 도보로 학교를 다니는 초등학교 학생이라고 가정하면, 이 이동 객체는 집에서 나와서 학교까지 이동하고, 학교에서는 이동이 없다가 다시 집으로 오고, 집 주위 놀이터나 학원으로 이동할 것이다. 그러면 이동할 때는 갱신이 필요하지만 학교, 집, 학원, 놀이터에 있을 때는 갱신이 필요 없다. 이런 객체를 추적할 때는 효율적인 방법이다. 이와 반대로 속도가 빠르고 이동이 많은 객체, 즉 자동차와 같은 객체를 추적할 때에는 짧은 시간에 오차 임계를 넘어 서기 때문에 잦은 갱신이 필요함으로 이 방법으로 위치 추정하는 것은 비효율적이다.

2.2 벡터 기반 위치 추적 방법

벡터 기반 위치 추적 방법은 이동 객체가 일정한 속도와 방향으로 이동한다고 가정한다. 이 방법의 경우 점 기반 방법과 달리 속도 정보가 추가적으로 필요하다.

이 방법의 알고리즘을 살펴 보면, 초기에 이동객체는 최근 받은 두 점을 기반으로 속도 벡터를 계산한다. 초기 위치와 속도 벡터를 이동 객체와 서버에 저장한다. 이동 객체에서는 이 정보를 기반으로 추정한 위치와 현재 위치 사이 거리를 계산하여 오차 임계 값보다 클 경우 다시 속도 벡터와 위치를 갱신하고, 서버에 갱신을 요청한다. 이 방법의 갱신은 저장된 위치와 속도 벡터로 이루어진 직선과 현재 위치의 거리 차가 오차 임계 값을 넘을 때 마나 발생한다. 서버에서는 갱신된 위치와 속도 벡터를 이용하여 이동 객체의 위치를 추정한다.

이 방법은 속도 벡터 정보가 정확할 경우 갱신할 필요 없이 완벽하게 추정할 수 있다. 비행기, 배, 고속도로를 달리는 자동차 등과 같은 이동 객체는 속도와 방향이 많이 변하지 않기 때문에 이 방법의 가정에 잘 만족한다. 이런 이동 객체들은 적은 갱신 횟수로 비교적 정확한 위치를 추정 할 수 있다. 이와 반대로 속도와 방향 정보가 많이 변하는 객체, 즉 꼬불꼬불한 시골길을 달리는 자동차, 축구 경기 중 공과 같은 이동 객체에 대해서는 갱신 횟수를 많이 줄일 수 없다.

2.3 도로 분할 기반 위치 추적 방법

도로 분할 기반 위치 추적 방법은 기본적으로 이동 객체는 도로 위에서만 움직인다고 가정한다. 이 방법의 경우 위해서는 앞에서 말한 2가지 방법과 달리 추가적으로 도로 정보가 필요하다.

이 방법의 알고리즘을 살펴 보면, 이동 객체가 처음 위치를 서버에 저장하고, 서버에게 자신이 속해 있는 도로에 대한 정보를 요청한다. 서버는 지도 정보를 검색하여 이동 객체가 속한 도로 정보를 보내준다. 이동 객체는 그 도로 위를 이동하며, 실제 위치와 추정한 위치를 비교하여 오차 임계를 넘을 경우나 도로 분할 정보가 끝났을 경우 서버에게 갱신을 요청한다. 특히 도로 분할 정보가 끝났을 경우에는 현재 이동 객체가 속한 위치의 도로 정보를 다시 요청한다. 이 방법의 갱신은 도로 정보가 끝나거나, 추정한 위치와 실제 위치가 오차 임계 값을 넘었을 때 발생한다. 서버에서는 이동 객체가 속한 도로에서 일정한 속도로 간다고 가정하고 이동객체의 위치를 추정한다.

이 방법은 이동객체가 도로 위에서만 움직일 때 적용 가능하므로 기차와 자동차 같은 이동 객체의 위치 추적에 좋은 성능을 발휘할 것이다. 기차 길과 고속도로는 분기점과 교차로 등이 적고, 움직이는 차량의 속도 변화도 빈번하지 않으므로 위치 정보 갱신 횟수를 줄일 수 있다. 반면 복잡한 도시의 도로를 달리는 자동차의 경우, 도로에 분기점이나 교차로들이 많고, 신호등이나 교통 정체와 같은 속도 변화 요인도 많으므로 갱신 횟수를 줄이기 어렵다.

3. Test Bed System을 이용한 비교

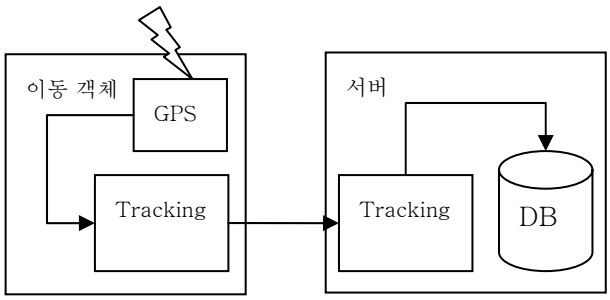

그림 1. Test Bed System의 구조

앞서 살펴본 3가지 알고리즘을 비교 평가하기 위하여 그림 1과 같이 Test Bed System을 구현하였다. 그림 1에서 GPS 모듈은 GPS 데이터 또는 GPS 로그 데이터를 이용하여 위치를 실시간으로 파악 한다. 서버와 이동 객체에 있는 Tracking 모듈은 앞서 언급한 위치 추적 방법들을 포함 하며, 서버의 데이터 베이스는 이동 객체의 갱신된 정보를 저장하고 서버와 이동 객체는 무선 네트워크로 연결되어 있고, 실험을 위해서 실제 GPS데이터를 사용하지 않고, 도로 네트워크 기반의 이동 객체를 생성하였다.

데이터는 T.Brinkhoff가 만든 도로 네트워크 기반의 이동 객체 생성기[2]를 이용하여 서울 도로 정보를 기반으로 이동 객체를 생성하였다. 생성된 이동 객체의 특징을 살펴 보면, 도로의 특성(고속도로, 골목길 등)에 따라 이동 객체의 속도가 다르고, 같은 도로 내에서도 속도가 변한다. 이동 경로는 시작점과 끝점을 지도에서 랜덤하게 생성하여, 도로의 특성을 고려하여 두 점을 연결하는 최단 경로를 찾아서, 그 경로로 이동하는 이동 객체의 궤적이 된다.

위와 같은 방식으로 생성된 이동 객체는 속도가 빠르고 이동 거리가 많으므로 점 기반 방식에 부적절하다. 따라서 본 논문에서는 점 기반 방법 제외하고, 벡터 기반 방법과 도로 분할 기반 방법에 대해 실험해 보았다. 본 연구에서 위치 추정 방법의 성능은 갱신 횟수와 추정한 위치 정확성으로 평가 하였다. 따라서 오차 임계 값을 5m부터 30m까지 5m간격으로 늘리면서 오차 평균과 갱신 횟수를 알아 보았다. 갱신 횟수는 실제 위치 정보 100개당 발생한 갱신 횟수를 말한다. 그리고 실험에 사용한 이동 객체들의 실제 GPS 신호의 길이는 평균 800번 정도이다. 아래 실험에서 나온 결과는 100개의 이동 객체들에 대한 평균값이다.

먼저 벡터 기반 방법을 살펴 보면, 그림 2와 같이 오차 임계 값이 증가 하면 갱신 횟수가 줄어들고, 오차 평균 값은 일정하게 증가한다. 따라서 벡터 방법은 임계 값을 조정함으로써 오차 평균 값과 갱신 횟수를 조절할 수 있다. 정확한 위치 정보를 얻고자 하면 오차 임계 값을 작게 하고, 갱신 횟수를 줄이고자 하면 임계 값을 크게 하면 된다.

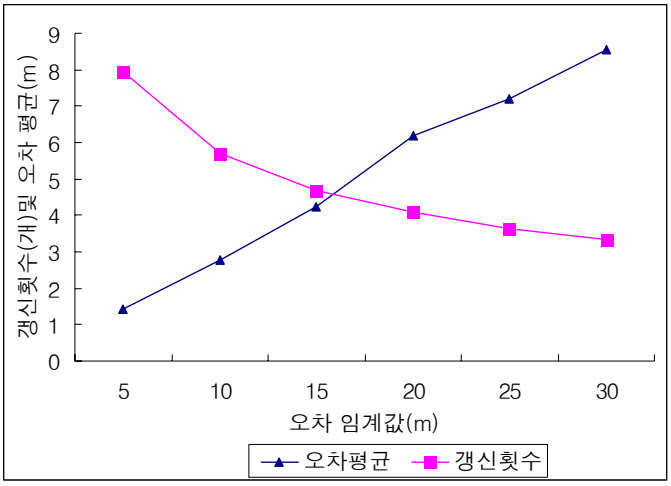

그림 2 벡터 기반 방법

도로 분할 기반 방법을 살펴 보면, 그림 3에서 보듯이 벡터 방법에 비하여 오차 평균 값의 변화가 적고, 갱신 횟수 변화가 거의 없고, 도로 분할로 발생하는 갱신 횟수는 오차 임계 값과 관계가 없다. 벡터 방법과 비교해보면, 갱신 횟수에서는 벡터 방법보다 좋지 않지만, 오차 평균은 이 방법이 좋다. 그림 3에서 보듯이 갱신 횟수에 중요하게 영향을 미치는 것은 도로 분할 상태이다. 도로 분할을 길게 만들 수 있다면, 갱신 횟수도 줄일 수 있을 것이다.

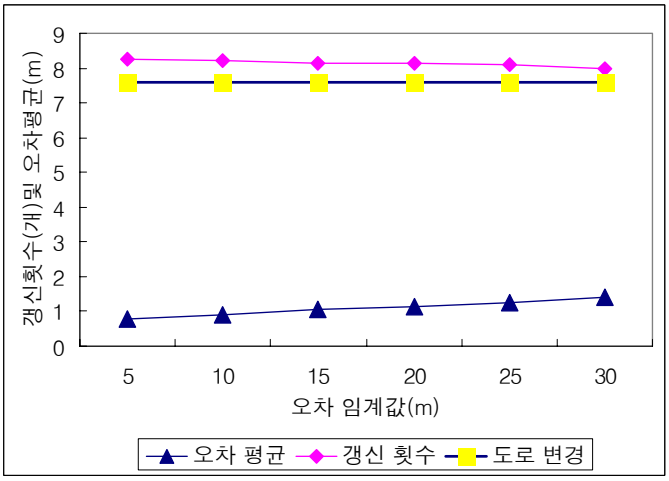

그림 3 도로 분할 기반 방법

4. 결론 및 향후 연구

현재까지 많은 위치 추적 방법들이 연구되고 발표되었지만, 실질적으로 실험을 통해 비교 할 수 있는 방법이 거의 없다. 본 논문에서 제시한 테스트 베드 시스템을 이용하면 여러 가지 위치 추적 방법을 실험할 수 있다. 테스트 베드 시스템을 이용하여 점 기반 방법과 벡터 기반 방법, 도로 분활 기반 방법의 위치 추정하는 것과 각 방법들의 장단점을 파악할 수 있었다. 이를 통해 얻은 정보를 바탕으로 더욱 좋은 위치 추적 방법을 만들 수 있다. 벡터 기반 방법은 오차 임계 값을 조정함으로써 사용자 요구에 조건에 충족하는 위치 추적 방법으로 만들 수 있고, 도로 분할 기반 방법은 도로의 길이를 길게 하여 갱신 횟수를 줄여 개선된 위치 추적 방법을 만들 수 있다.

Acknowledgment

본 논문은 한국 산업기술 재단 지역전력사업 석/박사 연구인력 양성사업의 연구 결과입니다.

참고 문헌

[1] M-Track. The M-Track project Web Site. http://www.cs.auc.dk/research/DP/mtrack/index.html

[2] T. Brinkhoff. Generating Network-Based Moving Objects. Scientific and Statistical Data Management, pp. 253-255, 2000

[3] C. S. Jensen, J. Kolar, T. B. Pedersen, and I. Timko, Nearest Neighbor Queries in Road Networks, in Proceedings of the Eleventh International Symposium on Advances in Geographic Information Systems, pp. 1-8. 2003

[4] K. Y. Lam, O. Ulusoy, T. S. H. Lee, E. Chan, and G. Li. An Efficient Method for Generating Location Updates for Processing of Location-Dependent Continuous Queries. Database Systems for Advanced Applications, pp. 218-225, 2001

[5] O. Wolfson, S. Chamberlain, S. Dao, L. Jiang, and G. Mendez. Cost and Imprecision in Modeling the Position of Moving Objects. ICDE 98, pp. 588-596, 1998

[6] J. Schiller and A. Voisard, Morgan Kaufman. Database Aspects of Location-Based Services, in Location-Based Services, 22 pages