

Smallworld를 이용한 이동객체 관리 및 위치 예측기의 구현

정영진 배중철 안윤애 류근호
충북대학교 데이터베이스 연구실
{yjjeong, jcbae, yeahn, khryu}@dmlab.chungbuk.ac.kr

Implementation of Moving Objects Manager and Location Predictor using Smallworld

Young Jin Jung Jong Chul Bae Yun Ae Ahn Keun Ho Ryu
Dept. of Computer Science, Chungbuk National University

요 약

이동객체란 시간에 따라 객체의 공간속성이 연속적으로 변하는 객체를 말한다. 기존의 데이터베이스를 이용하여 이를 관리할 경우, 단순히 객체의 현재 위치정보만을 저장하므로 과거 및 미래의 위치정보를 제공하지 못하는 단점을 가진다. 따라서 이 논문에서 제안한 이동객체 관리 및 위치 예측기는 과거 이력질의 뿐만 아니라 샘플링되지 않은 객체의 위치를 보간법을 사용하여 계산한다. 그리고 객체의 행위정보를 저장하고 시간에 대한 함수를 사용하여 가까운 미래에 대한 이동객체의 향후위치를 사용자에게 제공하는 하는 특징을 가진다.

1. 서론

시간에 따라 객체의 공간속성이 연속적으로 변하는 객체를 이동객체(Moving Objects)[3]라 한다. 이런 이동객체를 관리하기 위해서는 시간에 따라 연속적으로 변하는 객체정보를 저장해야 하는데, 이를 실제 구현할 때는 얼마간의 시간간격마다 객체정보를 저장하는 이산적 모델[2]을 사용하게 된다. 그리고 이를 기존의 데이터베이스를 이용해서 이동객체의 위치정보를 저장하면, 단순히 저장된 현재 객체정보에 대한 검색만을 할 수 있다. 따라서 이 논문에서는 현재의 객체위치 뿐만 아니라 과거, 미래의 객체 위치 변화를 추적하기 위하여 이동객체 정보 관리기에 보간법을 이용하여 샘플링되지 않은 이동객체의 위치를 계산하는 기능과 시간에 대한 함수를 이용하여 가까운 미래의 향후 위치를 계산하는 기능을 추가하였다. 따라서 사용자가 이동객체를 관리할 때 과거, 현재, 그리고 미래의 위치 정보를 제공받고 이를 활용할 수 있게 하였다. 이와 같은 이동객체 관리 및 위치 예측기를 GIS 도구인 smallworld를 이용하여 구현하였다.

2. 관련연구

기존 데이터베이스를 이용하여 시간에 따라 위치나 영역같은 공간정보가 연속적으로 변하는 이동객체정보를 관리하게 되면, 단순히 저장된 정보에 대한 검색만을 할 수 있다. 이와 같은 단순질의 외에 미래의 위치에 대한 질의를 처리해 주기 위해 DOMINO(Database for Moving Object) Project[6]는 기존 데이터베이스 위에 이동객체를 관리하는 software 모델을 추가한 형태의 MOST(Moving Object Spatio-Temporal) Model을 제안했다. MOST 모델은 빈번하게 변하는 객체의 위치 등의 속성값을 저장하기 위해 명시적인 갱신이 없어도 시간이 지남에 따라 속성정보가 변하는 Dynamic attribute를 사용한다. 또 이동객체의 향후위치, 등에 대한 질의처리를 위해서 FTL(Future Temporal Logic) Language를 제공하고 이동객체의 현재와 미래에 대한 위치질의에 답하기 위해서 객체의 속도, 방향 등의 행위정보(Motion information)[6]를 저장하고, 이를 활용한 시간에 대한 함수를 사용함으로써 객체의 향후위치에 대한 질의를 처리한다.

CHOROCHRONOS[1] 연구의 Data types 기반 접근에서 Luca[2]는 연속적인 이동객체의 값을 저장하기 위해 slice 안에서의 이동객체의 시간에 대한 값을 simple 함수로 나타낼 수 있는 "slice" 라는 조각들로 분할하는 Sliced Representation 을 제안하였다. 그러나 이들은 구현되어 특정분야에 적용된 사례가 없으므로 실제로 활용하여 적용되기에는 미비한 점이 있다. 우리는 이러한 문제를 이 연구에서 해결한다.

3. 이동객체 관리 및 위치 예측기 설계

3.1 구조

이 논문에서 설계한 이동객체 관리 및 위치 예측기는 이동객체 데이터베이스, GIS 도구, 이동객체 입력기, 이동객체 연산 처리기, 연속적인 위치 처리기로 구성된다.

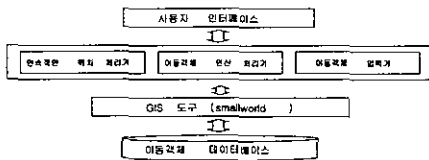


그림 1. 이동객체 관리 및 위치 예측기

그림 1은 이동객체 관리 및 위치 예측기의 전체구조를 나타낸다. 이동객체 데이터베이스는 현재정보 테이블, 과거정보 테이블로 구성되며, 이동객체 정보를 저장한다. 이동객체 입력기는 이동객체 정보를 입력받아 행위정보 처리후 데이터베이스에 저장한다. 이동객체 연산 처리기는 데이터베이스에 저장된 객체의 시간, 공간속성을 활용하여 시공간 연산을 한다. 연속적인 위치 처리기는 샘플링되지 않은 객체위치를 보간법을 이용하여 계산한다. GIS 도구는 공간객체를 저장, 관리한다. 이 논문에서는 Smallworld 공간관리 도구를 사용한다.

3.2 데이터베이스 구조

이 논문에서 다루는 이동객체는 이동 점만을 말하며, 차량 종류로 한정한다. 이동객체 정보 테이블은 이동객체의 현재, 과거 상태를 나타내는 "과거정보 테이블"과 "현재정보 테이블" 두 가지가 있다. 두 테이블의 속성정보는 같으며 객체의 행위정보를 저장하고, 시간에 대한 함수[5]를 사용하여 객체의 미래 위치를 계산한다.

표 1. 이동객체 정보 테이블 구조

OID	Name	ONO	Type	Speed	Orient	VT	X	Y
sys_id	Char(20)	char(4)	Int	Int	Int	Char(20)	Double	Double

표 1은 이동객체의 속성을 나타낸 것으로, 일반적 속

성과 이동방향 및 속도에 대한 행위정보, 그리고 시간정보가 있다. 행위정보에서 속도의 단위는 10m/m 이며, 방향은 16 방위를 사용하였다. 시간정보는 "년/월/일/시/분"의 정보를 가지며 "2000/10/10/13/10" 과 같이 10분 단위로 저장한다.

3.3 샘플링 사이의 이동객체 위치 결의처리 알고리즘

이동객체 정보가 샘플링되지 않은 객체 위치를 계산하기 위해 이 논문에서는 보간법을 사용하여 특정 시간에서의 이동객체 위치를 제공한다.

```

Algorithm InterpolationOfPositions(Object From_O, Object To_O,
                                   Double Time)
Input : Time, /* 위치를 구하려는 특정 시간 */
        From_O, To_O /* Time 의 앞뒤 시간대의 객체
Output : x, y /* Time 에서의 객체 위치 */
{
    /* Time 에 대한 보간법을 사용하여 x 좌표를 구한다. */
    x=From_O.x+ (To_O.x-From_O.x)(Time-From_O.vt)/(To_O.vt-From_O.vt)
    /* x 좌표에 대한 보간법을 사용하여 y 좌표를 구한다. */
    y=From_O.y+(To_O.y-From_O.y)(x-From_O.x)/(To_O.x-From_O.x)
    return x, y
}
    
```

그림 2. 연속적인 위치계산 알고리즘

그림 2는 선형보간법을 사용하여 연속적인 시간에 따른 이동객체의 위치를 구하는 알고리즘에 대해 나타낸 것이다. 정의한 특정시간과 그 시간의 바로 이전, 이후의 객체정보를 입력받아 샘플링되지 않은 객체위치를 초 단위까지 계산한다.

3.4 이동객체 향후위치 결의처리 알고리즘

현재정보 테이블에 저장된 정보를 바탕으로 시간에 대한 함수를 이용하여 가까운 미래 위치 결의를 계산한다.

```

Algorithm FutureLocation(Double x, Double y, Double dis, Int ori)
Input : x, y, /* 현재 정보 테이블에서 가져온 객체 위치 */
        dis, ori /* 거리 = 속도 * 시간 (행위정보를 이용) */
        /* 객체의 방향 */
Output : xx, yy /* 가까운 미래의 객체 위치 */
{
    /* 16 방위중 한 대표값으로 각도를 계산한다.
    angle = 2 * Math.PI * ori /16;
    xx = x + (int)( dis * Math.cos(angle)); /* x 좌표 */
    yy = y + (int)( dis * Math.sin(angle)); /* y 좌표 */
    return xx, yy
}
    
```

그림 3. 향후위치 계산 알고리즘

그림 3은 향후위치 계산 알고리즘을 나타낸 것이다.

입력값인 x, y 는 현재의 객체 좌표, dis 는 현재부터 질의한 미래시간까지의 시간차에 속도를 곱해서 움직이는 거리를 나타낸다. 그리고 ori 는 객체의 현재방향이다. 이런 입력값은 현재 정보 테이블에서 검색하여 가져오며, cos 과 sin 을 이용하여 가까운 미래의 위치를 계산한다.

4. 구현 및 실험 결과

이 논문에서 제안한 이동객체 관리 및 위치 예측기는 운영체제로 Windows NT 4.0을 사용하는 PC에서 GIS 도구인 smallworld 3.0의 공간 연산자의 시간적 확장과 smallworld 자체 데이터베이스를 사용하여 이동객체 정보 입력, 이동객체 검색, 이동객체사이의 거리계산, 샘플링되지 않은 객체의 위치계산, 이동객체의 향후 위치계산 기능을 구현하였다.

실험결과로는 렌트카 관리 서비스에 적용한 이동객체 관리 및 위치 예측기의 시공간 이동객체 관련 기능 중 시간간격에 따른 이동경로 검색 및 샘플링되지 않은 객체의 위치계산 기능에 대해 기술한다.

표 2. 15-Istana 의 이력정보

OID	Name	ONO	Type	Speed	Orient	VT	X	Y
6200	15-Istana	1000	1	20	14	2000/10/10/13/10	3321000	-233100
6207	15-Istana	1000	1	23	16	2000/10/10/13/20	3397000	-463600
6210	15-Istana	1000	1	24	1	2000/10/10/13/30	3668000	-516000
6213	15-Istana	1000	1	20	3	2000/10/10/13/40	3946000	-446300
6216	15-Istana	1000	1	24	2	2000/10/10/13/50	4022000	-344400

표 2는 2000년 10월 10일 오후 1시 10분부터 50분까지 과거 정보 테이블에 저장된 15-Istana의 이력정보이다.

질의 1) "2000년 10월 10일 오후 1시 14분 56초에 이동객체 15-Istana, board-15car, Sonata, Matize 에 대한 위치를 검색 하시오"

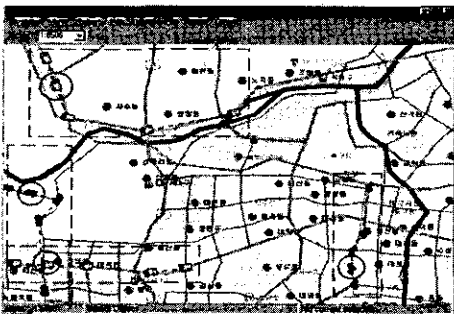


그림 4. 연속적 이동객체 위치 검색

그림 4는 질의 1의 실행결과이다. 점선 사각형은 2000년 10월 10일 오후 1시 10분부터 40분까지의 질의된 객체의 이동경로를 나타낸다. 점선사각형안의 객체들은 모두 같은 객체로서 시간에 따른 객체의 이동을 10분 간격

으로 나타내었고, 출력된 결과의 혼돈을 피하기 위해 같은 type의 객체는 같은 모양으로 출력하였다. 실선 원은 샘플링되지 않은 특정 시점에 대한 객체의 위치를 선형 보간법을 이용하여 계산한 것이다. 그리고, 계산된 객체는 저장된 정보가 없으므로 객체의 행위정보 또한 보간법을 사용하여 계산하였다.

표 3. 2000년 10월 10일 13시 14분 56초의 15-Istana의 정보

OID	Name	ONO	Type	Speed	Orient	VT	X	Y
6200	15-Istana	1000	1	21	15	2000/10/10/13/14/56	3356000	-338200

표 3은 데이터베이스에 저장된 정보를 바탕으로 선형보간법을 이용하여 계산한 2000년 10월 10일 오후 1시 14분 56초의 15-Istana의 정보를 나타내었고, 속도, 방향에 대한 행위정보 역시 보간법을 이용해 계산하였다.

5. 결론

이 논문에서는 이동객체 정보 관리기에 보간법을 이용하여 샘플링되지 않은 이동객체의 위치를 계산하는 기능과 시간에 대한 함수를 이용하여 가까운 미래의 향후 위치를 계산하는 기능을 추가한 이동객체 관리 및 위치 예측기를 구현하였다. 그리고, 시공간 연산의 효율을 높이기 위해 GIS 도구를 사용하였고 렌트카 관리 서비스에 적용하여 실제 실험함으로써 이동객체 관리에 유용하게 쓰일 수 있음을 확인하였다. 앞으로는 이동객체의 시공간적 불확실성을 좀더 좋은 알고리즘으로 처리하여 질의할 때 이동객체 정보의 정확성을 높이는 연구가 진행되어야겠다

6. 참고 문헌

- [1] Martin Erwig, Ralf Hartmut Guting, Markus Schneider and Michalis Vazirgiannis, "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Object in Databases", CHOROCHRONOS Technical Report CH-97-08, December 1997.
- [2] L.Forlizzi, R.H.Guting, E.Nardelli, and M.Schneider, "A Data Model and Data Structures for Moving Objects Databases", Proc. ACM SIGMOD Conf, Dallas, Texas, pp. 319-330, 2000.
- [3] Ralf Hartmut Guting, Michael H. Bohlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis, "A Foundation for Representing and Querying Moving Objects", CHOROCHRONOS, January 5, 2000.
- [4] Dieter Pfoser, Christian S. Jensen, "Capturing the Uncertainty of Moving-Object Representations", CHOROCHRONOS Technical Report CC-99-2, April 1999.
- [5] Simonsa Saltenis, Christian S. Jensen, Scott T. Leutenegger and Mario A. Lopez, "Indexing the Positions of Continuously Moving Objects", TIMECENTER TECHNICAL REPORT CH-99-03, October 1999.
- [6] Ouri Wolfson, Bo Xu, Sam Chamberlain and Liqin Jiang, "Moving Object Databases : Issues and Solutions", In Proc. Of the 10th Intl. Conference on Scientific and Statistical Database Management, Capri, Italy, 1998.