

시공간 이동 객체 연산자의 설계†

장승연⁰ 정영진 안윤애 류근호
충북대학교 데이터베이스 연구실
{syjang⁰, yijeong, yeah, khryu}@dblab.chungbuk.ac.kr

Design of Spatiotemporal Moving Object Operators

Seung Youn Jang⁰ Young Jin Jung Yoon Ae Ahn Keun Ho Ryu
Dept. of Computer Science, Chungbuk National University

요 약

최근 무선 환경에서 객체의 공간정보를 얻을 수 있는 기술들이 발전함에 따라 이동 객체를 대상으로 하는 여러 응용 시스템들이 등장하였다. 이런 응용들의 요구사항을 만족시키기 위해서는 연속적으로 변화하는 이동 객체의 공간 정보를 저장 및 관리 할 수 있어야 하며 다양한 사용자 질의를 처리해야 한다. 그러나 기존의 이동객체 데이터베이스에 대한 연구는 실세계의 사용자 질의 처리시 질의 대상이 되는 객체나 시간을 제한한다는 단점이 있다. 따라서 이 논문에서는 연속적으로 변화하는 이동 객체의 공간 정보를 관리하기 위한 데이터베이스 구조를 제시하고 이를 기반으로 사용자 질의 처리를 위한 시공간 이동 객체 연산자를 설계한다.

1. 서 론

기존의 데이터베이스는 시간과 공간을 독립적으로 취급하였거나 불연속한 시간상에서 공간 속성이 변화하는 시공간 객체를 다루어 왔기 때문에 실시간으로 변화하는 이동 객체의 공간 속성값을 저장 및 관리하고 이에 대한 사용자 질의 처리를 수행하기가 어렵다. 이로 인해 최근 이동 객체를 대상으로 하는 이동 객체 데이터베이스 시스템에 대한 연구[1,2,3,4]가 활발히 진행중이다.”

실세계 응용에서 “사고지역의 응급환자를 이송한 응급차A의 현재 이동 경로를 검색하라”, “현재 화재A지역의 화재피해면적 상태와 이 지역에 5분 이내에 도착할 것이라 기대되는 소방차를 검색하라”와 같은 이동 객체에 대한 사용자질의는 시간과 공간 제약을 모두 가지며 가까운 과거나 현재, 또는 미래 시간에 초점을 맞추는 특징을 지닌다. 그러나 대부분의 이동 객체데이터베이스에 대한 연구는 특정 어플리케이션 지향적이거나 추상 데이터 타입을 기반으로 연구가 이루어지고 있기 때문에 현실세계에서의 사용자 질의 처리시 질의 대상이 되는 객체나 시간을 제약하는 문제가 있다.

따라서 이 논문에서는 이동 객체의 연속적인 공간 변화 정보를 처리하기 위한 데이터베이스 구조를 제공하고 이를 기반으로 실세계에서의 다양한 사용자 질의를 처리하기 위한 연산자들을 설계한다.

2. 관련 연구

DOMINO(Database fOr MovINg Object) 프로젝트[1]에서는 현재 객체의 위치 정보에 대한 단순 질의 외에 미래 위치에 대한 질의를 처리해 주기 위해 기존 데이터베이스 위에 이동 객체를 관리하는 소프트웨어 모델을 추가한 형태의 MOST모형을 제안하고 있다. MOST 모형을 동적 속성을 통해 빈번하게 변화하는 객체의 위치

속성값을 표현하며 이동 객체의 향후위치 등에 대한 질의처리를 위해서 FTL언어를 제공한다. 그러나 이 모델의 경우 이동 점만을 대상으로 하며 이력 정보를 기록하지 않음으로 해서 객체의 과거 위치에 관한 질의가 가능하지 않다는 단점이 있다. CHOROCRONOS 연구 그룹에서는 이동 객체 표현을 위해 시간 흐름별로 분할된 “슬라이스(slice)” 라는 조각들을 통해 시간상에서 변화하는 이동 객체의 속성 정보를 표현하는 슬라이스 표현[2]을 제안하고 이를 바탕으로 연산자들을 정의하였다[3]. 그러나 추상 데이터타입을 기반으로 이동 객체 연산자들을 설계함으로써 데이터베이스로 관리되지 않는 시간 축 현재나 미래 시간을 대상으로 하는 사용자 질의는 다루기가 어렵다는 문제가 있다. 또한 시간에 대한 함수로서 이동 객체를 표현함으로써 질의처리시 복잡도가 증가하는 문제가 있다[5].

3. 이동 객체 데이터베이스 구조

3.1 시공간 이동 객체 유형

이동 객체는 시간에 따라 객체의 공간 정보가 연속적으로 변경되는 객체로서 시간상에서 위치정보가 연속적으로 변화하는 이동 점(예: 사람, 자동차)과 위치정보 외에 모양정보도 변화하는 이동 영역(예: 화재, 오염지역)으로 나눌 수 있다[6]. 이 논문에서는 시간속성과 공간속성을 동시에 가지는 이동 객체를 표현하기 위해 표 1과 같은 속성들을 정의하였다.

표 1. 이동 객체 표현을 위한 속성

타입	속성	의미
시간	유효시간	객체 속성 값이 참값을 가질 때의 시간 시간 구간 사용 (가정: 거래시간 = 유효시간)
	포인트	2차원 평면상에서 x,y 좌표값으로 표현되는 점
공간	심플라인	시작점과 끝점 두 개의 점으로 이루어진 선
	폴리라인	점들의 리스트로 구성된 선
	폴리곤	점들의 리스트로 구성된 영역

† 이 연구는 한국과학재단 KRC(성주대정보통신연구센터)의 연구비 지원으로 수행되었음

표 1은 이동 객체를 표현하고 이동 객체 관련 연산 설계시 이용될 속성 값들에 대해서 정의한 것이다.

이동 점은 일정한 경로를 따라 움직이는 객체를 대상으로 하며 일정 시간 간격별로 샘플링 된 유클리드 평면상의 점으로 표현한다. 이동 영역은 일정 시간 간격별로 샘플링 된 유클리드 평면상의 폴리곤으로 표현하며 객체의 모양이 확대 및 축소하는 것을 허용한다.

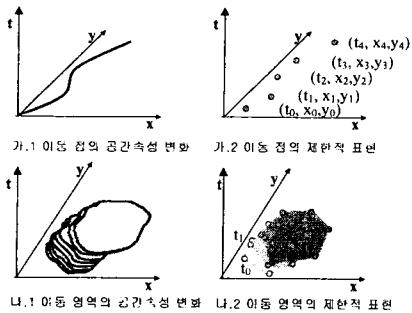


그림 1. 시공간 이동 객체의 표현

그림 1은 일정 시간 간격별로 이동 객체의 공간 속성 값을 샘플링하여 제한적으로 표현한 것을 나타낸다. 이때 샘플링 되지 않은 공간 정보값들은 샘플링 된 공간 정보값에 적절한 위치추정함수를 적용함으로써 이동 객체의 연속적인 공간 속성을 표현해 준다. 이동 영역은 위치뿐만 아니라 모양까지 변화하므로 이 논문에서는 이를 위해 Shumilov의 Key-Frame 보간법[7]을 응용한다. 이 방법은 Key-Frame 보간법의 저장공간 문제를 해결하기 위해 샘플링 시점마다 변화정보를 확인하고 변화된 정보를 시간 구간별로 저장하는 방법을 사용하였다. 그리고 축소 및 확대되는 경우에 따라 모양변화정보 추정을 위해 사용되는 매핑구조를 다르게 적용함으로써 연속적으로 변화하는 이동 영역의 모양정보를 추정해 준다.

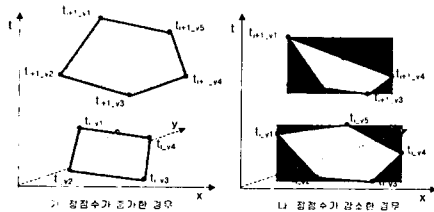


그림 2. 이동 영역의 모양 변화

그림 2는 이동 영역의 모양 변화를 나타낸 것으로 t_{i+1} 시점에서 샘플링된 이동 영역의 정점 정보를 t_i 시점과 비교하여 모양이 확대, 축소되는 경우 변화된 모양을 추정할 수 있는 정점 정보들을 별도로 저장한다. 이 정보들을 Poly_ch라 한다면 모양이 확대한 경우 Poly_ch와 t_{i+1} 시점에서의 폴리곤의 각 정점에 적절한 위치추정함수를 적용하고 축소된 경우에는 Poly_ch와 t_i 시점에서의 정점 정보에 위치추정함수를 적용함으로써 객체의 변화하는 모양 및 위치 정보를 추정한다.

3.2 이동 객체 정보 저장 스키마

이동 객체 연산 설계시 고려하는 이동 객체의 공간 정보는 다음과 같은 구조로 표현된다.

표 2. 일반속성 스키마

MO_ID	MO_type	MO_name
oid	int	string

표 2는 이동 객체의 시공간 속성 외에 객체 아이디나 객체 이름, 객체 타입 등과 같은 일반 속성정보를 저장 관리하는 스키마이다.

표 3. 이동 점의 과거 및 현재 정보 스키마

MP_ID	Vfrom	Vto	PointF	PointT
oid	date	date	point	point

표 3은 이동 점에 대한 과거나 현재, 또는 가까운 미래 질의를 처리해 주기 위한 스키마로 과거 정보와 현재 정보를 별도로 분리하여 저장한다. 이동 점의 이력정보가 저장되는 과거정보 스키마는 과거시간에 대한 사용자 질의처리를 사용하며 각 객체마다 가장 최근에 샘플링 된 두 시점의 위치 정보를 저장, 관리하는 현재정보스키마는 현재나 가까운 미래에 대한 질의 처리시 필요한 백터 정보를 추정하기 위해 사용한다.

표 4. 이동 영역의 과거 및 현재 정보 스키마

MR_ID	Vfrom	Vto	PolyF	PolyT	Ch_check	PolyCh
oid	date	date	polygon	polygon	int	polygon

표 4는 이동 영역의 공간정보 저장을 위한 스키마로 이동 점과 마찬가지로 두 개의 스키마로 관리되며 객체의 모양변화정보를 확인해주는 Ch_check속성과 변화된 모양정보를 저장하는 PolyCh속성을 추가로 가진다.

4. 시공간 이동 객체 연산자

4.1 연산자의 종류

3절에서 소개한 이동 객체 유형과 저장 스키마를 기반으로 실제세계에서의 사용자 질의 처리를 위해 필요한 이동 객체 연산자들을 표 5와 같이 설계하였다. 이때 사용되는 표기 MP는 이동 점을, MR은 이동 영역을 말하며 Tp는 시점을 Ti는 시간 구간을 말하고 id는 객체의 아이디를 나타낸다.

표 5. 이동 객체 연산자

대상	연산자	표기 (입력값 → 출력값)
MP	MPTraj	Ti, id → trajectory
	MPLength	Ti, id → length
	MPFTime	Tp, id, location → Tp, time
	MPatVel	Tp, id → Tp, average velocity
MR	MRTrav	Ti, id → traverse
	MRArea	Tp, id → Tp, area
	MRPerimeter	Tp, id → Tp, perimeter
	MRatCenter	Tp, id → Tp, location
MP, MR	MOatTime	Tp, id → Tp, location
기타	ProjTime	Tp, Value → Time
	ProjValue	Tp, Value → Value
	CalMin	[Tp, Value] → Tp, MinValue
	CalMax	[Tp, Value] → Tp, MaxValue

표 5는 이동 객체에 대한 질의 처리시 사용되는 시공간 연산자들 중에서 이동 객체만을 대상으로 하는 연산

자들로 실제 사용자 질의를 처리하기 위해서는 표 5에서 정의된 이동 객체 연산자 외에 기존의 시간 및 공간 연산자들이 필요하다. 표 5에서 MPTraj는 특정 시간구간 동안 이동 점이 이동한 궤적을 반환하며 MPFTime은 이동 점이 특정 위치까지 도달하는데 걸리는 시간값을 계산한다. MRTrav는 이동 영역이 이동한 모습을 반환하며 MOatTime은 특정 시점에서의 이동 객체의 위치값을 제공한다. ProjValue는 시점값과 특정값을 입력으로 받아서 특정 값만을 반환하는 역할을 하고 CalMin은 특정값들을 입력으로 받아서 최소값을 반환한다.

표 5에서 설계된 연산자들 중에서 특정 시점에서 이동 객체의 위치값을 반환하는 MOatTime의 알고리즘은 표 6과 같다.

표 6. MOatTime 연산 알고리즘

```

Function MOatTime(Tp, id)
input : Timepoint, object id
output : Timepoint, Location
MOatTime(Tp, id)
{
    timepoint tp Tp;
    location loc;
    int MOId=CheckMO(id); //객체 유형 확인
    int CT;
    if(MOId <= 0) //이동 점의 경우
    {
        //질의 시간 및 질의 시점에서의 객체 관리여부 확인
        switch(CT=CheckTime(tp))
        case 0: //과거 질의
            if(CheckEx(tp, id)) //질의 시점에서의 객체 존재 여부 확인
            {
                loc SearchLocation(tp, id);
            }
            else //객체가 존재하지 않는 경우
            {
                loc UncertainF(tp, id); //과거 위치추정 함수 이용
                break;
            }
        case 1: //현재 및 미래 질의
            if(CheckEx(tp, id)) //현재 시간이 저장되어 있는지 확인
            {
                loc SearchLocation(tp, id);
            }
            else
            {
                loc UncertainF(tp, id); // 미래 위치 추정 함수 이용
                break;
            }
        case 2: //질의시간이 데이터베이스로 관리되지 않는 경우
            exception process
            break;
        return loc, tp;
    }
    if(MOId > 1) //이동 영역의 경우
    {
        //이동 점의 경우와 같음
        return loc, tp;
    }
}
    
```

표 6에서 이동 영역은 이동 점에 적용했던 방법을 이동 영역을 구성하는 각 시점에서의 점점들에 적용함으로써 객체의 위치를 추정한다.

4.2 질의 표현

설계된 연산자를 기반으로 실제계의 사용자 질의를 표현해 보면 다음과 같다. 이때 예제 질의에서는 다음과 같은 오염지역과 응급차량, 병원이라는 추상 스키마를 사용하였다.

PollArea (id:string, name:string, geo:moving region)
 EmTruck (id:string, name:string, geo:moving point)
 Hospital (id:string, name:string, geo:point)

“현재 응급차A의 위치에서 반경 3km 이내에 존재하는

병원을 검색하라”라는 이동 점에 대한 사용자 질의는 다음과 같은 구문을 통해 처리될 수 있다.

```

Select h.name
From Hospital as h, EmTruck
Where Distance(h.geo,ProjValue(MOatTime(current,'응급차A')))<3;
    
```

현재 시간에서의 응급차 위치는 MOatTime과 ProjValue연산을 통해 얻어지며 기존의 공간 연산자 Distance를 통해 사용자 질의를 처리해 준다.

“현재 A오염의 피해지역과 피해면적을 구하여라”라는 이동 영역에 대한 사용자 질의는 다음처럼 표현한다.

```

Select ProjValue(MOatTime(current,'A오염')),
ProjValue(MRArea(current,'A오염')) From PollArea;
    
```

질의처리시 현재 시간에서의 A오염 면적을 얻기 위해 MRArea연산을 사용한다.

5. 결론

시간의 변화에 따라 연속적으로 이동하는 객체를 대상으로 하는 응용 시스템들의 요구사항을 처리하기 위해서는 연속적으로 변화하는 이동 객체의 공간속성을 표현해 주어야 하며 다양한 사용자 질의를 처리해야 한다. 그러나, 기존의 이동 객체에 대한 연구는 사용자 질의 처리시 질의 대상이나 질의 시간을 제한하는 문제가 있다. 이와 같은 문제를 해결하기 위해 이 논문에서는 이동 객체의 연속적인 공간정보를 표현하기 위한 데이터베이스 구조를 제시하고 이를 바탕으로 사용자 질의 처리를 위해 필요한 연산자들을 설계하였다. 향후 연구로는 다양한 사용자 질의 처리를 위해 추가적으로 필요한 연산자들을 설계하고 설계한 연산자들을 구현하는 작업이 남아 있다. 또한 실제 응용 시스템에 이 연산자들을 적용 및 평가하는 작업이 수행될 것이다.

참고문헌

- [1] A.P.Sistla, Ouri Wolfson, S.Chamberlain, S.Dao, "Modeling and Querying Moving Objects", ICDE, pp. 422~432, 1997.
- [2] L.Forlizzi, R.H.Guting, E.Nardelli, and M.Schneider, "A Data Model and Data Structures for Moving Objects Databases", Proc. of ACM SIGMOD Conf, pp. 319~330, 2000.
- [3] R.H.Guting, M.H.Bohlen, M.Erwig, C.S.Jensen, N.A.Lorentzos, M.Schneider, and M.Vazirgiannis, "A Foundation for Representing and Querying Moving Objects", ACM Transactions on Database Systems, Vol.25, No.1, pp.1~42, 2000.
- [4] 신기수, 안윤애, 배종철, 정영진, 류근호, "GIS를 이용한 시공간 이동 객체 관리 시스템", 한국정보처리학회 논문지, 제 8-D권, 제2호, pp. 105~116, 2001.
- [5] S.Grumbach, P.Rigaux and L.Segoufin, "Spatio-Temporal Data Handling with Constraint", ACM International Workshop on Advances in Geographic Information Systems, 1998.
- [6] M.Erwig, R.H.Guting, M.Schneider and M. Vazirgiannis, "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Object in Databases", GeoInformatica. Vol.3, No.3, pp.269~296, 1999.
- [7] S.Shumilov, J.Siebeck, "Database Support for Temporal 3D data Extending the GeoToolkit", EC-GI&GIS Workshop, 2001.