

GIS를 이용한 이동객체 관리기의 설계 및 구현

정영진, 배종철, 안윤애, 류근호
충북대학교 데이터베이스 연구실

e-mail:{yjjeong, jcbae, yeahn, khryu}@dmlab.chungbuk.ac.kr

Design and Implement for Moving Object Manager Using GIS

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

요약

이동객체란 시간에 따라 객체의 공간정보가 연속적으로 변경되는 객체이다. 이와 같은 이동객체를 관리하는 이동객체 관리기는 일반적으로 시공간 데이터베이스 형태로 구축할 수 있다. 기존의 시공간 데이터베이스는 저장된 정보에 대한 단순질의만을 처리하지만, 이 논문에서 제안한 이동객체 관리기는 데이터베이스에 저장된 객체의 위치정보 검색 뿐만 아니라 향후 객체가 이동할 위치 정보를 사용자에게 미리 제공한다. 이 논문에서는 이와 같은 이동객체 정보 저장과 미래정보 제공을 위한 이동객체 관리기를 설계 및 구현하였다.

1. 서론

시간에 따라 위치나 모양 등의 공간적 속성이 변하는 택시, 유조선, 비행기 등 실세계의 이동객체들을 관리하려면 객체의 시간속성과 공간속성을 모두 관리하는 시공간 데이터베이스를 이용한 이동객체 관리기가 필요하다. 기존의 시공간 데이터베이스는 저장된 정보에 대한 단순질의만을 처리하지만, 여기에 향후 객체의 이동할 위치 정보까지 제공한다면 사용자는 이동객체에 대한 더 많은 정보를 활용할 수 있을 것이다. 이 논문에서는 그 동안 연구된 시공간 이동객체 관리 데이터베이스와 시공간 연산자를 바탕으로 하는 GIS 기반 이동객체 관리기의 모델을 제안한다. 이동객체를 효율적으로 관리하기 위해 GIS 도구를 사용하면 GIS 도구의 공간연산자에 시간차원을 확장한 시공간 연산자를 구현하는 것이 용이해진다. 따라서, 이와 같은 이동객체 정보 저장과 미래정보 제공을 위한 이동객체 관리기를 설계 및 구현한다. 이와같은 이동객체 관리기의 응용으로는 항법 시스템(Navigational System), 워 게임 모델(War Game Model) 등이 있다.

2. 관련연구

이동객체는 시간에 따라 객체의 공간정보가 연속적으로 변경되는 객체로써 크게 이동점(Moving Point)과 이동영역(Moving Region)으로 나눌 수 있다. 이동점은 시간에 따라 객체의 위치(Position or Location)가 변하는 것이고, 이동 영역은 시간에 따라 객체의 위치뿐만 아니라 모양(Shape)까지 변하는 것이다[1].

이동객체의 위치는 시간에 따라 연속적으로 변화한다. 이러한 이동객체의 모델링[2]에는 연속적 모델과 이산적 모델방법이 있다. 연속적 모델은 이동객체를 무한(Infinte)한 점들의 집합으로 표현하는 것으로 3차원 공간상에서 연속적인 곡선(Continuous Curve)으로 표현할 수 있다. 이러한 연속적 모델은 이동객체의 이동정보를 정확하게 묘사할 수 있지만 무한한 점들을 컴퓨터에 저장하고 관리하는 것이 어렵기 때문에 구현에 있어 적합치 않은 반면, 이산적 모델은 이동객체를 유한(Finite)한 점들의 집합으로 표현하는 것으로 3차원 공간상에서 다중선(Polyline)으로 표현할 수 있다. 이는 이동객체의 이동정보를

근사치로써 보여주는 것으로 구현이 가능하다.

이동객체를 관리하려면 시간에 따라 변화하는 공간 객체들의 저장 및 관리를 위한 시공간 데이터베이스의 구축 및 시공간 연산이 필요하다. 시간 데이터베이스나 공간 데이터베이스를 시공간 데이터베이스로 확장하는 방법으로는 시간 데이터베이스에 공간 데이터베이스를 확장하는 방법과 공간 데이터베이스에 시간 데이터베이스를 확장하는 방법이 있다.

기존의 DBMS는 시간에 따라 위치가 변화하는 이동객체를 저장하기 위해 빈번한 수정연산이 요구되어 이를 극복할 수 있는 방법이 필요하다. 또한, 이동객체의 현재 상태만을 기록하므로 향후 이동위치를 계산할 수 없는 단점이 있다.

이 논문에서는 GIS 도구에서 지원하는 공간연산자를 시간적으로 확장하는 방법을 택하고, 기존의 DBMS에서 저장된 정보만을 보여주던 것에 향후 이동위치까지 계산하는 기능을 더하여 사용자에게 제공한다.

3. GIS 기반 이동객체 관리기의 설계

3.1 구조

이 논문에서 설계한 이동객체 관리기는 이동객체 데이터베이스, GIS 도구, 이동객체 입력기, 이동객체 검색기, 이동객체 연산기로 구성된다.

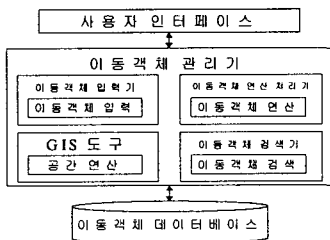


그림 1. 이동객체 관리기 구조

그림 1은 이동객체 관리기의 전체구조를 나타낸다. 이동객체를 위한 시공간 데이터베이스는 현재 정보 테이블, 과거 정보 테이블로 구성된다. 이동객체 입력기는 이동객체 정보를 입력하고, 이동객체 연산처리기는 데이터베이스에 저장된 객체의 시간, 공간 속성을 활용하여 시공간 연산을 한다. 이동객체 검색기는 원하는 이동객체 정보를 찾아준다. GIS 도구는 공간객체를 관리하며 이 논문에서는 Geowin 공간관리 도구를 사용한다.

3.2 이동객체 데이터베이스 구조

이동객체 정보 테이블은 이동객체의 현재, 과거 상태를 나타내는 2개의 테이블로 구성되며 표 1에서와 같이 각각의 객체별 속성과 위치, 날짜, 이동 방향 및 속도에 대한 정보를 가지고 있다. 이동정보 입력에 따라 방향 및 속도가 변경되면, 현재 정보를 수정한다.

표 1. 이동정보 테이블의 구조

필드명	Type	의미
OID	OID	객체 구분자
Name	String	객체 이름
Group	int	소속 그룹 번호
Type	int	객체 구분 번호
x	int	x 좌표
y	int	y 좌표
Speed	int	이동 속도
Orient	int	이동 방향
VTs	String	유효 시작 시간
VTc	String	유효 종료 시간

표 1은 이동객체에 대한 속성을 나타낸 것이다. 속도단위는 10 Km/h 이고, 방향은 그림 2의 16방위를 사용한다.

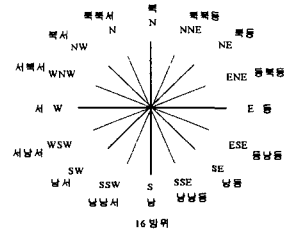


그림 2. 16 방위

그림 2의 실선들은 16 방위에 대한 각각의 대표값을 나타낸 것이다.

3.3 GIS 도구

이 논문에서 사용한 GIS 도구인 Geowin은 Java 클래스들로 구현된 Internet/Intranet 환경에서 작동하는 GIS Client/Server 시스템으로 속성 정보 테이블과 표 2와 표 3과 같이 공간 인덱싱 테이블, 공간 속성 테이블로 구성된다. 속성 정보 테이블은 공간 객체의 속성정보를 저장하기 위한 테이블로 속성으로 OID를 갖고, 나머지 속성은 추가시킬 수 있다.

표 2. 공간 인덱싱 테이블

NID	PID	SID	HGT	OID	XMIN	YMIN	YMAX	XMAX
-----	-----	-----	-----	-----	------	------	------	------

표 2는 공간 인덱싱 테이블로 빠른 공간 데이터 조회 및 공간검색을 위한 테이블이다. 공간상의 객체들에 대한 R*-Tree 인덱스를 생성하여 저장한다.

표 3. 공간 속성 테이블

OID	ELEMENTS	EID	GEOTYPE	SEGMENTS	SID
POINTS	X0	Y0	X1	Y1	X2
	Y2	X3	Y3	X4	X5
	Y5	X6	Y6	X7	X8
	Y8	X9	Y9		

표 3 은 공간 정보 테이블로 공간상에 존재하는 공간 객체들의 실제 좌표 정보를 저장하는 테이블이다. 하나의 레코드는 10개의 점을 가질 수 있고, 하나의 공간 객체는 복수개의 레코드로 구성될 수 있다. X0, Y0 필드는 필수 입력 필드이다.

3.4 이동객체 연산처리기

이동객체 연산처리기는 이동객체 관리기의 가장 중심역할을 하는 처리기로써 이동객체 연산자를 처리한다. 이동객체 연산자는 이동객체 종류와 적용분야에 따라 다양한 연산자들이 사용된다.

표 4. 이동객체 연산자

연산자	기능
GetDistanceAtTime	특정 시간에 이동점들 간의 거리값 구한다.
GetOrient	한 점에서, 다른 점으로의 방향을 구한다.
GetSpeed	한 점에서, 다른 점으로의 속도값 구한다.
PositionAtTime	특정 시간에 이동점의 위치값 구한다.
GetTrejectory	주어진 이동점의 궤적을 구하여 선(Line)으로 표현한다.
GetLength	주어진 선의 길이값 구한다.

표 4 는 Erwig[1] 이 제시한 이동객체 연산자들을 바탕으로 본 논문에서 설계 구현된 이동객체 연산자들의 종류를 나타낸 것이다.

이동객체 연산자를 기반으로 이동객체를 입력 및 검색하는 등의 사용자가 수행 가능한 기능을 설명하면 다음과 같다.

- ① 이동객체 정보 입력 기능 : 객체속성, 위치 및 유효시간을 입력한다. 속력 및 방향은 입력하지 않고 입력한 이동객체에 대한 과거정보가 있을 때, 그에 대한 속력과 방향을 계산하여 저장한다. 과거정보가 없을 때는 속력과 방향 값은 "null" 이 된다. 연산자는 GetOrient, GetSpeed를 사용한다.
- ② 시간별 객체 위치 검색기능 : 주어진 시간에 객체의 위치정보를 가져온다. 미래일 경우, 속도와 방향을 참고하여 미래의 위치를 계산한다. 연산자는 PositionAtTime를 사용한다.
- ③ 객체 이동거리 계산 기능 : 한 객체가 특정 시간 간격 동안 이동한 거리를 계산한다. 연산자는 GetTrejectory, GetLength를 사용한다.
- ④ 객체간 거리 계산 기능 : 특정시간에 두 객체의 거리를 계산한다. 미래일 경우, 속도와 방향을 참

고하여 미래에서의 객체간 거리를 계산한다. 연산자는 GetDistanceAtTime를 사용한다.

4. 구현

4.1 구현환경

이동객체 관리기는 운영체제로 Windows NT 4.0 을 사용하는 PC에서 Java 1.2.2 으로 Oracle 7 데이터 베이스와 Geowin 공간관리 시스템을 사용하여 이동객체 정보 입력, 시간별 객체 위치 검색, 객체 이동거리 계산, 객체간 거리 계산 기능을 구현하였다.

4.2 실험

관리가 이동객체를 어떻게 관리하는지 알아보기 위해 표 5 에서처럼 이동객체 정보를 입력받아 데이터베이스에 저장했다.

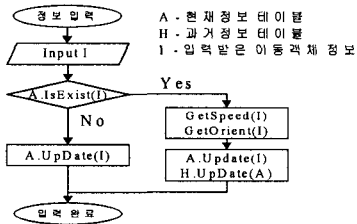


그림 3. 이동객체 정보 입력 순서도

그림 3 은 이동객체 입력 과정을 순서도로 나타낸 것이다. 그림 4 와 같은 이동정보 입력창으로 표 5 와 같은 데이터를 이동객체 데이터베이스에 입력한다.

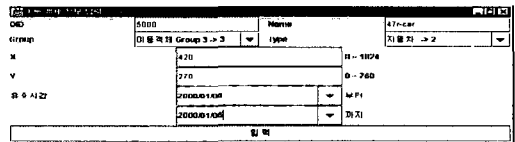


그림 4. 이동정보 입력

그림 4 는 2000/01/06일 47r-car 의 이동정보를 입력하는 화면이다.

표 5. 이동객체 정보 입력

OID	Name	Group	Type	x	y	VTs	VTe
5000	47r-car	3	2	420	270	2000/01/06	2000/01/06
5000	47r-car	3	2	427	310	2000/01/07	2000/01/07
5000	47r-car	3	2	442	360	2000/01/08	2000/01/08

표 5 는 실험시에 입력한 객체정보이다. Speed와 Orient는 입력할 때 넣는 것이 아니라, 저장된 정보를 참고하여 계산되는 것이기 때문에 입력할 때는 넣지 않는다.

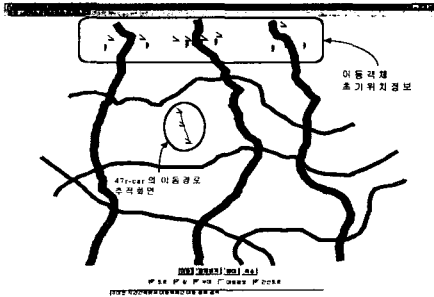


그림 5. 이동경로 추적

그림 5 는 입력된 이동객체 47r-car 의 이동경로 추적 화면으로써 이동객체가 어떤 경로로 움직였는지를 나타내준다. 이동객체 정보를 처음 입력할 때는 Speed(속도), Orient (방향) 를 계산할 근거가 없다. 그러므로, 두 속성값은 “null” 값을 가진다. 처음 입력하는 것이 아니라면, 이전 정보와 비교하여, Speed 와 Orient 값을 계산하여 객체정보 테이블에 이동객체 정보를 저장한다. 그림 5 와 같은 결과가 나올 때 실제 데이터베이스에 저장된 정보는 표 6 과 표 7 과 같다.

표 6. 과거정보 테이블

OID	Name	Group	Type	x	y	Speed	Orient	VTs	VTe
5000	47r-car	3	2	420	270			2000/01/06	2000/01/06
5000	47r-car	3	2	427	310	17	8	2000/01/07	2000/01/07

표 6 은 표 5 의 객체정보를 입력한 후의 과거정보 테이블로써 47r-car 의 과거정보를 저장한다.

표 7. 현재정보 테이블

OID	Name	Group	Type	x	y	Speed	Orient	VTs	VTe
5000	47r-car	3	2	442	360	21	7	2000/01/08	2000/01/08

표 7 은 표 5 의 객체정보를 입력한 후의 현재정보 테이블로써 47r-car 의 현재 위치, 속도, 방향을 나타낸다.

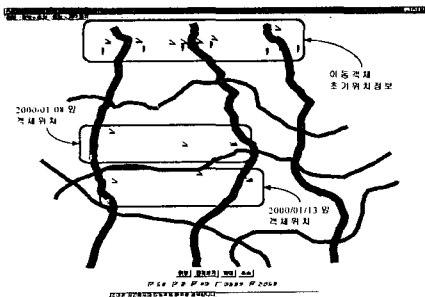


그림 6. 시간별 이동객체 위치

그림 6 은 데이터베이스에 저장된 객체들의 이동

정보가 '2000/01/10' 일 이전 정보만 저장된 상태에서, '2000/01/08' 일과 '2000/01/13' 일의 객체 위치를 나타낸 것이다.

그림 6 의 시간에 따른 이동객체 위치를 보면, '2000/01/08' 일은 데이터베이스에 저장된 객체 위치 정보를 그대로 나타낸 반면, '2000/01/13' 일은 앞으로의 이동위치를 데이터베이스에 저장된 이동객체 정보를 바탕으로 계산한 것이다. 그림 5 와 그림 6 을 보면 이동객체 연산자가 관리기에서 어떻게 쓰이는지 확인할 수 있다.

5. 결론

이 논문에서는 이동객체 정보 저장과 미래정보 제공을 위한 이동객체 관리기를 설계 및 구현하였다. 그리고, 실제세계의 이동객체들을 관리하기 위해 GIS 기반에서 이동정보를 입력하고, 시간에 따른 객체의 위치, 이동경로, 객체간 거리를 검색하고, 객체의 이동거리를 계산할 수 있도록 이동객체 관리기를 구성하였다. 아울러, 실제 실험을 함으로써 이동객체 관리에 유용하게 쓰일 수 있음을 확인하였다. 앞으로는 이동객체의 시간적 불확실성을 처리하여 질의 시 이동객체 정보의 정확성을 높이는 연구가 진행되어야겠다.

참고문헌

- [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, 2000, 319-330
- [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] K.Ryu, "A Research on Temporal and Spatial Reasoning and Uncertainty Processing". Final Report, Spatiotemporal Reasoning Project. Chungbuk National University, July, 2000.
- [5] S.Dutta, "Topological Constraints : A Representational Framework For Approximate Spatial And Temporal Reasoning". Second International Symposium on Large Spatial Databases, SSD'91, 1991
- [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". CHOROCHRONOS project, Dec., 1997.