

GIS를 이용한 시공간 이동 객체 관리 시스템

신 기 수^{*} · 안 윤 애^{**} · 배 종 철^{***} · 정 영 진^{***} · 류 근 호^{****}

요 약

이동 객체는 시간에 따라 공간 객체의 위치 및 영역이 연속적으로 변경되는 시공간 데이터이다. 기존의 데이터베이스 시스템을 이용하여 시공간 이동 객체를 관리할 경우 다음의 두 가지 문제점을 가진다. 첫째, 시간에 따라 변화되는 위치 정보에 대한 빈번한 갱신이 발생된다. 둘째, 항상 객체의 현재 상태만이 저장되므로 시공간 이동 객체의 과거와 미래에 관한 정보를 제공하지 못한다. 따라서, 이 논문에서는 빈번한 갱신 없이 이동 객체의 이력 정보를 관리할 뿐만 아니라 과거, 현재, 그리고 가까운 미래에 관한 모든 위치 정보를 제공할 수 있는 시공간 이동 객체 관리 시스템을 제안한다. 제안 시스템에서 이동 객체 정보는 위치를 나타내는 위치 정보와 이동 습성을 나타내는 행위 정보로 구분된다. 특히, 행위 정보의 변경 처리 알고리즘을 사용하여 최소한의 이력 정보만으로도 모든 객체의 위치 정보를 검색할 수 있는 방법을 제안한다. 그리고, 제안한 방법을 전장분석 시스템에 적용하여 구현하였으며, 이를 통해 관계형 데이터베이스와 GIS 시스템을 이용하여 실제계의 시공간 이동 객체의 과거, 현재 및 가까운 미래의 위치 정보를 관리할 수 있음을 알 수 있었다.

A Spatiotemporal Moving Objects Management System using GIS

Keysoo Shin^{*} · Yunae Ahn^{**} · Jongchul Bae^{***} ·
Youngjin Jung^{***} · Keunho Ryu^{****}

ABSTRACT

Moving objects are spatiotemporal data that location and shape of spatial objects are changed continuously over time. If spatiotemporal moving objects are managed by conventional database systems, moving objects management systems have two problems as follows. First, update for location information changed over time is occurred frequently. Second, past and future information of moving objects are not provided by systems because only current state of objects is stored in the system. Therefore, in this paper, we propose a spatiotemporal moving objects management system which is able to not only manage historical information of moving objects without frequent update, but also provide all location information about past, current, and near future. In the proposed system, information of moving objects are divided into location information for representing location and motion information for representing moving habits. Especially, we propose the method which can search location information of all objects by use of changing process algorithms with minimum history information. Finally, we applied the proposed method to battlefield analysis system, as the result of experiment, we knew that past, current, and near future location information for moving objects are managed by relational database and GIS system.

키워드 : 이동 객체(Moving Object), 시공간 데이터(Spatiotemporal Data), 위치 정보(Location Information), 행위 정보(Location Information)

1. 서 론

현실 세계에서 움직이는 모든 객체들은 시간 및 공간 속성을 동시에 가지며 시간에 따라 공간 속성이 변하는 특성을 가진다. 이러한 데이터를 시공간 데이터라고 하는데, 특히 시공간 데이터 중에서도 시간에 따라 공간상에서 위치나 모양이 변하며 이동하는 데이터를 시공간 이동 객체(Spatiotemporal Moving Objects)라 한다[1]. 시공간 이동 객체를 이용한 응용 분야에는 선박 관리 시스템, 운송 관리

시스템, 교통 관리 시스템, 그리고 전장 분석 시스템 등이 있으며, 이들 응용에서는 시간에 따라 변하는 객체의 공간 정보를 효과적으로 표현하고 관리할 수 있는 방법이 요구된다[2].

그런데, 기존의 데이터베이스 시스템을 이용하여 시공간 이동 객체를 저장 및 관리할 경우에는 연속적인 갱신(Update) 연산이 발생되며, 시공간 이동 객체의 현재 상태에 대한 정보만을 유지하게 되므로, 이동 객체의 미래 정보는 물론 과거에 대한 위치 정보 역시 제공하지 못하는 단점을 가지게 된다. 이것을 해결하기 위해 DOMINO(Database for Moving Object) Project[3]와 CHOROCHRONOS[4] 연구에서 데이터 모델 지향적인 접근 방법론이 제시되었지만, 이들은 시공간 이동 객체의 일반적인 특징들에 대한 연구일 뿐이며 적용 영역에 따른 이론 정립과 그 적

* 이 연구는 과학재단 특정기초연구비(#1999-2-303-006-3) 지원과 KAIST 해킹 바이러스 연구센터의 연구비 지원으로 수행되었음.

† 정 회 원 : 한국통신 연구개발본부 가입자망연구소

** 준 회 원 : 충북대학교 대학원 전자계산학과

*** 정 회 원 : 충북대학교 대학원 전자계산학과

**** 종신회원 : 충북대학교 컴퓨터학과 교수

논문접수 : 2000년 11월 17일, 심사완료 : 2001년 3월 14일

용 및 구현 사례가 없으므로 실제 응용 시스템에 적용하기에는 미비한 점이 많다.

따라서, 이 논문에서는 시간에 따라 연속적으로 변하는 시공간 이동 객체의 이산적인 위치 정보 표현 방법, 시공간 이동 객체의 위치를 검색 할 수 있는 시간 종속적 함수, 그리고 이동 정보의 효율적인 이력 관리 기능을 가지는 GIS 기반 시공간 이동 객체 관리 시스템을 개발한다. 개발 시스템에서 이동 객체의 정보 표현은 이동 위치를 나타내는 위치 정보와 이동 습성을 나타내는 행위 정보로 구분된다. 특히, 이 논문에서는 행위 정보의 변경 처리 알고리즘을 사용하여 최소한의 이력 정보만으로도 모든 객체의 위치 정보를 검색할 수 있는 방법을 제안하며, 시간 종속적 함수를 이용하여 이동 정보의 과거, 현재, 그리고 가까운 미래에 관한 모든 위치 정보 검색을 가능하게 한다.

그리고, 이 논문에서는 시공간 이동 객체 관련 응용 분야의 하나인 전장분석 시스템에 적용하여, 시간에 따라 변하는 행위 정보의 표현, 이동 객체의 위치 정보 모델링, 그리고 이동 객체 연산자를 설계 및 구현함으로써 제안 시스템이 실제 응용에 적용 가능함을 보인다. 이 시스템은 관계형 데이터베이스 시스템인 오라클(Oracle) DBMS에서 시공간 이동 객체를 표현하고, 시공간 이동 객체의 과거, 현재 및 향후 이동 위치를 검색할 수 있는 시공간 이동 객체 데이터 모델을 설계한다. 또한 Geowin 공간 관리 도구와 Java 언어를 이용하여 구현한다.

이 논문의 구성은 다음과 같다. 2장에서는 논문의 전반적인 이해를 돕기 위해 시공간 데이터 모델 및 연산자, 시공간 이동 객체에 대해 서술한다. 3장에서는 이 논문에서 제안하는 시공간 이동 객체 정보 표현을 위한 위치 정보 및 행위 정보의 관리 방법에 대해 기술한다. 4장에서는 GIS 기반 시공간 이동 객체 관리 시스템을 제안하고, 이를 전장 분석 시스템에 적용한 데이터 모델 및 알고리즘을 제시한다. 5장에서는 구현 시스템의 환경 및 실험 내용을 보이고, 마지막으로 6장에서는 결론 및 향후 연구과제를 제시한다.

2 관련연구

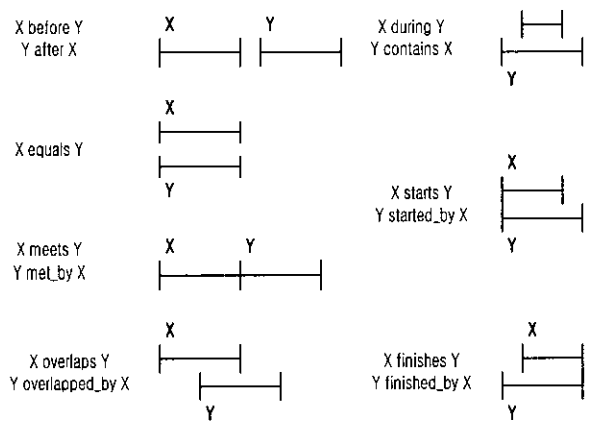
시공간 이동 객체 관리 시스템의 관련연구로 시공간 데이터 모델과 연산자, 그리고 시공간 이동 객체에 대해 지금까지 연구된 내용들을 소개하고 문제점을 파악한다.

2.1 시공간 데이터 모델과 연산자

시공간 데이터[4]는 시간에 따라 공간적 특성이 변하는 객체들을 나타낸다. 이러한 시공간 데이터를 관리하기 위해서는 시공간 데이터베이스의 구축 및 시공간 연산자의 제공이 필요하다.

먼저, 시공간 데이터베이스 구축을 위한 시공간 데이터 모델에는 실제계에서 발생하는 데이터에 대한 시간 적용과 그 범주에 따라 스냅샷 데이터 모델[5], 시공간 복합 데이터 모델[6], 사건 지향 모델[7], 이력 그래프 모델[8-10], 3 영역 모델[11], 객체지향 모델[12] 등이 연구되었다. 이들은 모두 시공간 데이터 모델링 하는 방법이지만, 각각 나름대로의 장점과 단점을 가지고 있다. 스냅샷 모델은 쉽고 단순하지만 자료 중복에 대한 문제[7]가 있고 시공간 복합 데이터 모델은 자료 중복의 문제를 어느 정도 해결하였지만 동일한 이력을 갖는 공간 객체의 구분이 불가능하다[13]. 또한 사건 지향 또는 사건 기반 데이터 모델에서는 발생 이력들이 기본 도면에 링크드 리스트로 연결되어 있기 때문에 최근 상태를 얻기 위해서는 기본 도면으로부터의 변화 상태를 조합해야 한다는 단점을 가진다. 시공간 객체 지향 모델은 시공간 객체를 쉽게 표현할 수 있지만 구현이 복잡하고 어렵다는 문제점이 있다. 그러므로 시공간 데이터 모델은 응용에 적합한 모델을 선택하여야 더 큰 효율을 기대할 수 있다.

다음으로, 시공간 연산자는 저장된 시공간 데이터를 조작할 수 있는 기능을 제공하는 것을 말한다. 지금까지 연구된 시공간 연산은 주어진 시간 위상관계 연산자를 만족하는 유효 시간과 거래 시간을 가지는 객체들에 대해 공간 연산을 수행하는 것으로, 크게 시간 위상관계 연산자와 공간 위상관계 연산자로 분류할 수 있다. 시간 위상관계 연산자는 대상이 되는 객체들의 시간 값 사이의 위상 관계에 대한 연산으로 Allen[14]이 제안한 13가지 연산자가 널리 사용되고 있으며 연산의 종류와 의미는 (그림 1)과 같다.



(그림 1) Allen의 시간 위상관계 연산자

공간 위상관계 연산자는 공간 관리 시스템에 따라 서로 다른 기능들이 제공되고 있고 대표적인 연산들로는 <표 1>에 요약한 바와 같이 *Enclosure*, *Identification*, *Inside*, *Intersection*, *Overlap*, *Proximity* 등이 있다.

<표 1> 공간 위상관계 연산자

연 산 자	기 능
Enclosure	포함 관계
Identification	동지 관계
Inside	특정 영역 내에 위치
Intersection	교차 관계
Overlap	겹침 관계
Proximity	인접 관계

2.2 시공간 이동 객체

시공간 이동 객체[4, 15]는 시간에 따라 객체의 공간 정보가 연속적으로 변경되는 객체로 크게 이동 점(Moving Point)과 이동 영역(Moving Region)으로 나눌 수 있다. 이동 점은 시간에 따라 객체의 위치(Position or Location)가 변하는 것으로, 이동 점 객체의 예로는 사람이나 동물, 자동차, 비행기, 배 등이 있다. 그리고, 이동 점 객체는 시간에 따라 위치가 변하며 특정 시점에 대한 질의는 그 시간에 이동 점 객체가 존재하는 위치를 표현하는 점(Point)을 반환한다. 이동 점에 대한 질의는 시공간 이동 객체의 위치에 대한 질의이며 시공간 이동 객체가 가지는 면적(Area)에 대해서는 질의를 수행하지 않는다. 이동 점과 관련한 질의들은 객체의 이동 경로나 방향, 거리등에 대한 질의들로써 <표 2>와 같다.

<표 2> 이동점 관련 질의

이동 점	질 의
사 람	A 지역을 순찰중인 경찰관의 위치는?
동 물	우리 나라에 서식하는 겨울 철새인 가마우지의 이동 경로를 보이시오.
차	승객이 요구한 위치에서 가장 가까운 택시는?
배	1시간 내에 부산항에 도착 예정인 선박은?

이동 영역은 시간에 따라 객체의 위치뿐만 아니라 모양(Shape)까지 변하는 것으로 한 국가의 행정 구역이나 폭풍의 영향권, 암세포의 상태 등을 예로 들 수 있다[4]. 이러한 이동 영역 객체는 시간에 따라 위치와 모양이 변하며 특정 시점에 대한 질의는 그 시점에 이동 영역이 존재하는 위치와 모양을 표현하는 영역(Region)을 반환한다. 이동 영역에 대한 질의는 특정 시간에 시공간 이동 객체의 위치나 모양에 대한 질의일 수 있으며 이와 관련한 질의들은 <표 3>과 같다. 이동 영역에 대한 질의는 이동 점과 마찬가지로 이동 경로나 방향, 거리등에 대한 질의를 포함하지만 추가적으로 이동 객체의 모양에 대한 질의가 포함된다.

<표 3> 이동영역 관련 질의

이동 영역	질 의
국 가	고구려의 영토가 가장 넓었을 때의 국가 경계는 어떤 모습이었는가?
폭 풍	2000년 6월 우리나라에 영향을 끼쳤던 4호 태풍 '카이탁'의 이동 경로 및 영향 지역을 보여라.

시공간 이동 객체를 표현하기 위한 데이터 모델에서는 연속적으로 변하는 시공간 이동 객체의 위치 정보를 표현하기 위한 특별한 방법이 요구된다. 이러한 시공간 이동 객체의 위치는 시간에 따라 연속적으로 변하기 때문에 시공간 이동 객체를 모델링 할때 연속적 모델과 이산적 모델로 구분된다. 연속적 모델은 시공간 이동 객체를 무한(Infinte)한 점들의 집합으로 표현하는 것으로 3차원 공간상에서 연속적인 곡선(Continuous Curve)으로 표현할 수 있다. 반면, 이산적 모델은 시공간 이동 객체를 유한한 점들의 집합으로 표현하는 것으로 3차원 공간상에서 폴리라인(Polyline)으로 표현할 수 있다. 이는 시공간 이동 객체의 이동 정보를 근사값으로 보여주는 것으로 구현이 가능하다. (그림 2)는 연속적 모델과 이산적 모델의 표현 방식을 나타낸 것이다.



(a) 연속적 표현 (b) 이산적 표현

(그림 2) 이동 객체의 표현

시공간 이동 객체를 표현하기 위한 새로운 데이터 모델의 필요성은 기존의 데이터베이스 모델로 시공간 이동 객체를 직접 표현할 수 없다는 것을 의미한다. 기존의 데이터베이스로 시공간 이동 객체를 직접 표현할 경우 시공간 이동 객체의 위치가 시간에 따라 계속적으로 변하므로 이를 데이터베이스에 기록하기 위해 빈번한 갱신 연산이 요구되고, 기존의 데이터베이스는 시공간 이동 객체의 현재 상태만을 기록하므로 시공간 이동 객체의 과거 및 향후 이동 위치와 관련된 질의에 적절히 응답할 수 없는 문제점들이 발생된다.

이와 같은 문제점들을 해결하기 위해 Sistla는[16]에서 시공간 이동 객체의 속성을 동적 속성(Dynamic Attribute)과 정적 속성(Static Attribute)으로 분류하여 관리하는 방법을 제시하였다. 동적 속성은 명시적 수정 연산 없이 시간에 따라 값이 변하는 속성을 의미하며, 특정 시간에 동적 속성의 값을 구할 수 있는 시간 종속적 함수를 제공한다. 정적 속성은 시간이 지나도 값이 변하지 않는 속성으로 일반적인 속성들을 나타낸다. 또한, Wolfson은[3]에서 객체의 이동 습성에 대한 정보 유지의 필요성을 제시하였다. 그러나, 지금까지의 연구에서는 이동 객체의 개념적인 모델링 및 이론적인 방법들만 제시되었고 실제 응용분야에 적용하여 구현된 사례가 거의 없다.

따라서, 이 논문에서는 지금까지 연구된 이동 객체의 이동론들을 바탕으로 이동 객체의 과거, 현재, 그리고 가까운 미래의 위치 정보를 검색할 수 있는 이동 객체 관리 시스템을 제시한다. 또한, 제안 시스템의 적용가능성을 보이기 위해 전장분석을 위한 응용 시스템을 개발하였다.

3 시공간 이동 객체 표현

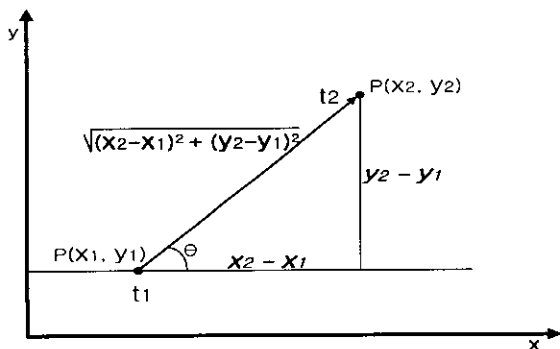
시공간 이동 객체에 대한 향후 이동 위치 검색을 위해서는 첫째, 이동 객체가 동일한 행위 정보를 가진 상태로 이동 중일 때의 시작 위치를 나타내는 위치 정보가 필요하다. 둘째, 이동 객체의 이동 습성을 나타내는 행위 정보가 필요하며, 셋째, 위치 계산을 위한 이동 함수에 관한 정보가 필요하다. 이 장에서는 위치 정보 및 행위 정보의 표현 방법에 대해 기술한다.

3.1 위치 정보의 표현

이 논문에서 제시하는 시공간 이동 객체 관리 시스템의 대상이 되는 이동 객체는 이동 점 객체로만 한정시키고, 이동 점 객체의 위치 정보 표현 방법에 대해 서술한다.

향후 이동 위치 검색 질의는 설정된 위치 정보에 행위 정보를 매개변수로 사용하는 이동 함수를 적용하여 수행된다. 이동 함수는 현재까지 입력된 이동 정보를 기반으로 객체의 행위 정보와 행위 정보의 적용 시작 위치인 위치 정보를 이용하여 객체의 향후 이동 위치를 계산하기 위한 함수들이다.

먼저, 이동 함수에 대해 살펴보자. 시간 t_1 과 t_2 사이에 시공간 이동 객체가 (x_1, y_1) 에서 (x_2, y_2) 로 이동하였다고 하면 이동 거리와 이동 방향 θ 는 (그림 3)과 같이 표현할 수 있다.



(그림 3) t_1 과 t_2 사이에 이동 객체의 위치 변화

(그림 3)에서 삼각함수와 삼각형의 정의를 이용하여 시공간 이동 객체의 이동 방향을 구할 수 있다. 다음의 식 (1)에서 각 θ 는 동쪽을 기준으로 하는 이동 방향이 된다.

$$\sin\theta = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad (1)$$

행위 정보에서 객체의 시작 위치를 구할 수 있고, 이동 정보 입력은 객체의 현재 위치를 포함하므로 위의 식 (1)에 의해 이동 방향을 구할 수 있다. 각 θ 의 범위가 0° 에서 360° 라고 할 때 $\sin\theta$ 의 값은 \sin 그래프에서 볼 수 있듯이 두 개가 나오므로 (x_1, y_1) 과 (x_2, y_2) 의 좌표를 검사하여 (x_1, y_1) 을 기준으로 (x_2, y_2) 가 몇 사분면에 존재하는지를 파악해야 한다.

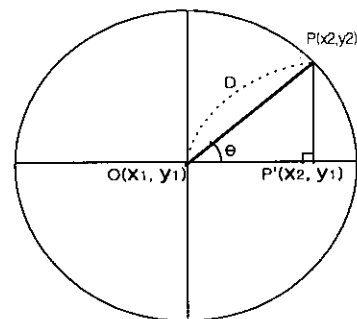
시공간 이동 객체의 이동 속도는 행위 정보에 설정된 이동 속도를 새로 입력된 이동 정보의 이동 속도와 비교한다. 속도 변화가 정의된 오차의 범위 이내이면 명시적 수정 연산 없이 이동 정보를 저장한다. 두 점 사이의 이동 거리 D 는 다음 식 (2)를 이용하여 구할 수 있다.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

이동 속도는 t_1 시간부터 t_2 시간까지 이동한 거리 D 를 시간 $|t_2 - t_1|$ 으로 나누면 구할 수 있다. 그러므로 이동 속도 S 는 식 (2)와 식 (3)으로부터 구할 수 있다.

$$S = \frac{D}{|t_2 - t_1|} \quad (3)$$

식 (1)과 식 (3)에 의해 시공간 이동 객체의 이동 방향과 속도가 계산된다. 이를 기반으로 시공간 이동 객체의 이동 위치를 검색하기 위해 원의 방정식을 이용한다.



(그림 4) 원의 방정식

(그림 4)에서 $O(x_1, y_1)$ 은 동일한 행위 정보를 가지는 이동 객체의 시작위치이고, $P(x_2, y_2)$ 는 검색하려는 위치가 된다. 주어진 시간 이후에 이동 가능한 객체의 위치는 식 (2)에서 구해진 거리 D 와 시간을 곱한 거리를 반지름으로 하는 원이 된다. 이 중에서 식 (1)에서 구해진 이동 방향 θ 에 해당되는 지점을 구하면 객체의 향후 이동 위치를 구할 수 있다.

원의 방정식에서 중점으로부터 주어진 각 θ 와 반지름 D 를 가지는 점 $P(x_2, y_2)$ 의 좌표는 식 (4)와 (5)에 의해 구할 수 있다. 여기서 반지름 D 는 식 (2)의 거리 D 와 동일하다.

$$x_2 = D \cos \theta \quad (4)$$

$$y_2 = D \sin \theta \quad (5)$$

지금까지 설명한 방법으로 시공간 이동 객체의 향후 위치를 검색할 수 있다. 하지만 이러한 시간 종속적 함수의 정확성을 높이기 위해서는 행위 정보를 유지하는 것이 필요하다.

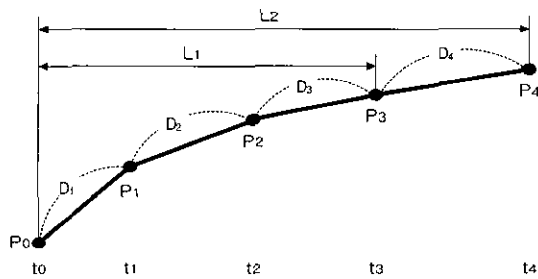
3.2 행위 정보의 표현

이 논문에서는 이동 객체의 행위 정보로 속도와 방향에 관한 정보를 사용한다. 그리고, 시공간 이동 객체는 시간 t_0 에서 t_4 로 흐르는 동안 객체의 위치 P 가 P_0 에서 P_4 로 이동하며, 시간의 단위는 모두 동일하게 사용됨을 가정한다.

3.2.1 속도 정보의 표현

속도 정보는 이동 정보의 입력 시에 계산된 속도를 검사하여 행위 정보에 기록된 속도 정보와의 차이가 주어진 오차의 범위를 넘을 경우 명시적 갱신 연산을 수행한다. 이 논문에서는 속도 변화를 판단하는 방법으로 다음의 두 가지 방법을 사용한다.

첫 번째 방법은 속도의 평균을 구하여 속도 변화를 판단하는 방법이다.



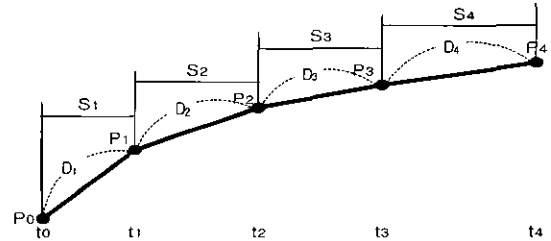
(그림 5) 평균 속도

(그림 5)에서 n 번째 이동 정보인 P_n 의 평균 이동 속도 S_n 은 다음 식 (6)으로 구할 수 있다.

$$S_n = \frac{\sum_{i=1}^n D_i}{t_n - t_0} \quad (6)$$

행위 정보의 변경은 이동 정보 입력 시점에서의 평균 속도 S_n 과 행위 정보에 기록된 속도 정보 사이의 차이가 주어진 오차의 범위를 벗어나면 수행되며, 행위 정보가 변경된 경우에는 그 시점의 평균 이동 속도로 행위 정보를 갱신한다.

두 번째 방법은 입력 단계별 속도를 파악하여 속도 변화를 판단하는 방법이다.



(그림 6) 입력 단계별 속도

(그림 6)과 같이 행위 정보의 변경은 행위 정보에 기록된 속도 정보와 현재 속도 사이의 차이가 주어진 오차의 범위를 넘어서는 경우 수행된다. 예를 들어 P_4 의 입력에서 이동 속도의 변경 여부를 파악하는 것은 P_4 의 이동 속도 S_4 와 행위 정보에 기록된 속도 정보 사이의 차이를 비교하는 것이다. (그림 6)의 경우 시간의 단위는 모두 동일하므로 이동 거리 D_4 가 P_4 의 속도가 되며, P_4 의 입력에서 행위 정보가 변경된 경우, 행위 정보의 속도 정보에는 S_4 가 설정된다.

앞에 서술한 두 가지 방법을 비교해 보면, 첫 번째 방법은 이동 정보의 속도 평균을 구하여 속도 정보의 변화를 파악하므로 간헐적인 속도 변화에 쉽게 영향을 받지 않는다. 즉, 속도의 변화가 전체 이동 정보로 분산되어 지속적인 속도 변화가 발생하거나 전체 평균 속도에 영향을 줄만큼 큰 차이의 속도 변화가 발생해야 행위 정보의 변경이 수행된다. 반면에 두 번째 방법은 입력된 이동 정보의 이동 속도만으로 속도 정보의 변화를 파악하므로 속도의 변화에 민감하여 첫 번째 방법에 비해 빈번한 갱신이 발생한다.

3.2.2 방향 정보의 표현

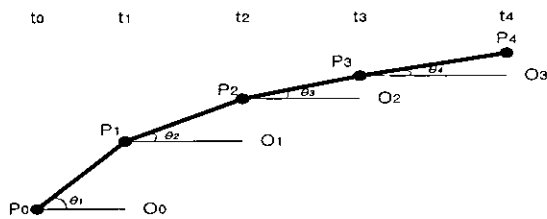
방향 정보의 변경은 기본적으로 시공간 이동 객체의 이동에 따라 이동 방향을 나타내는 각 θ 가 변경되고, 이 각 θ 가 주어진 오차의 범위를 벗어날 경우 행위 정보의 변경으로 보고 명시적 갱신 연산을 수행하는 것이다. 점 P 는 시공간 이동 객체의 이동 위치를 나타내며, O 는 기준선을, θ 는 이동 방향을 나타낸다. O 의 X좌표는 P 의 X좌표이며 O 의 Y좌표는 기준점의 Y좌표이다. 각 θ 는 명시적 갱신이 수행되는 시점에 설정된 기준 각이다. 또한 θ_n 은 기준선 O (또는 O_n)와 이동 객체의 위치 P_n 사이의 각을 나타내며 θ_1 에서 θ_4 로 변한다고 가정한다. 두 방법 모두 동쪽을 0° 로 한다. 방향 정보의 변화를 판단하는 방법으로 이 논문에서는 두 가지 방법을 사용한다.

첫 번째 방법은 n 번째 이동 정보의 입력에서 시공간 이동 객체의 이동 방향을 나타내는 각 θ_n 은 n 번째 이동 정보의 위치와 $n-1$ 번째 이동 정보의 위치 그리고 기준선

O_{n-1} 이 이루는 각을 구하여 이를 행위 정보에 기록된 방향 정보와 비교하는 것에 의해 이동 방향의 변경 여부를 판단한다. 이 방법에서는 기준선이 동적으로 구해지므로 각각의 이동 정보 입력마다 새로운 기준선 O_{n-1} 의 위치를 계산하는 과정이 필요하다. n 번째 이동 정보의 이동 방향을 나타내는 각 θ_n 은 식 (7)로 표현할 수 있다.

$$\theta_n = \angle P_n P_{n-1} O_{n-1} \text{ (단, } n \geq 1 \text{)} \quad (7)$$

다음의 (그림 7)에서 이동 정보 P_1 이 입력될 때 행위 정보에 기록된 방향 정보인 θ 와 현재 입력하는 이동 정보의 이동 방향을 나타내는 θ_1 의 차이를 계산하여 주어진 오차의 범위를 넘으면 행위 정보를 변경한다. P_2 역시 θ 와 θ_2 의 차이로 행위 정보 변경 여부를 판단하며 이후는 모두 동일하다.



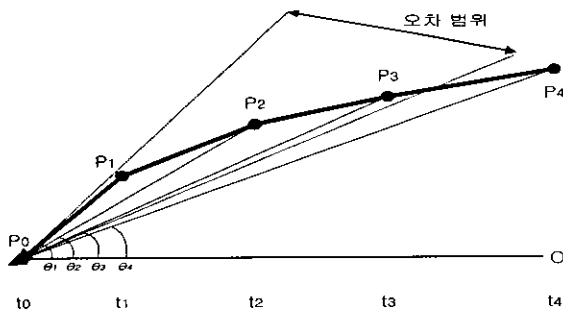
(그림 7) 동적 기준선

만일 P_4 의 입력 시점에 행위 정보가 변경되었다고 가정하면, 행위 정보의 위치 정보에는 P_4 , 방향 정보에는 θ_4 가 설정된다.

두 번째 방법은 행위 정보의 명시적 수정 연산 수행시 설정된 시작 위치와 입력된 이동 정보의 위치사이의 각을 기준으로 이동 방향의 변화를 판단한다. 이 방법에서 기준선 O 는 항상 고정적이다. n 번째 이동 정보의 이동 방향을 나타내는 각 θ_n 은 식 (8)로 표현할 수 있다.

$$\theta_n = \angle P_n P_0 O \text{ (단, } n \geq 1 \text{)} \quad (8)$$

다음의 (그림 8)에서 점선은 16방위 중의 한 부분으로 동일한 방향으로 처리되는 각의 범위, 즉, 오차의 범위이다. θ_n 은 P_n 의 이동 방향을 나타낸다.



(그림 8) 고정 기준선

(그림 8)에서는 이동 방향을 나타내는 각 θ_4 가 오차의 범위를 벗어나는 시점인 t_4 의 입력에서 행위 정보의 변경이 발생한다. 이때 행위 정보의 위치 정보에는 P_4 의 위치가, 방향 정보에는 P_4 가 위치하는 영역을 나타내는 방위의 대표 값이 설정된다.

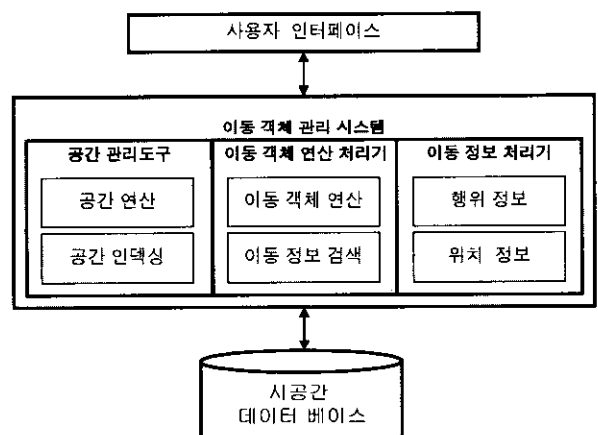
두 가지 방법 모두 각각의 이동 정보 입력에서 시공간 이동 객체의 이동 방향을 나타내는 θ_n 을 구해야 한다. 그러나, θ_n 을 이루는 세 점의 특성은 두 방법에 차이가 있다. θ_n 을 $\angle P_1 P_2 P_3$ 라고 하면, 첫 번째 방법은 P_1, P_2, P_3 모두가 계속적으로 변경된다. 즉 이들 세 점을 구하기 위한 추가적인 연산이 필요하다. 반면에 두 번째 방법은 P_2 와 P_3 는 고정적이며 P_1 만을 구하면 되므로 첫 번째 방법보다 추가적인 연산이 적다. 두 방법에서 행위 정보로 기록되는 방향 정보의 값을 살펴보면 첫 번째 방법에서는 특정 크기의 θ 를 가지며 오차의 크기는 응용 프로그램에서 정의되고, 두 번째 방법은 해당 방위별 대표 값이 저장된다. 두 방법은 응용의 특성에 따라 선택 가능하다.

4 이동 객체 관리 시스템

이 장에서는 GIS를 이용한 시공간 이동 객체 관리 시스템의 구조를 제시하고, 이를 특정 응용에 적용한 데이터베이스의 구조, 이동 객체 연산자, 그리고 행위 정보 처리 알고리즘을 제시한다.

4.1 시스템 구조

이 논문에서 제시한 시공간 이동 객체 관리 시스템은 관계형 데이터베이스 시스템을 기초로 시간과 공간 데이터를 처리하도록 확장하여 이동 객체를 표현하고 관리한다. 시간 차원의 확장을 위해 유효 시간을 지원하는 속성을 추가하고, 공간 차원의 확장을 위해서는 공간 관리 도구의 기능을 일부 사용한다. 이러한 GIS 기반 시공간 이동 객체 관리 시스템의 구조는 다음과 같다.



(그림 9) 시공간 이동 객체 관리 시스템 구성도

(그림 9)는 시공간 이동 객체 관리 시스템의 전체 구성을 나타낸 것이다. 사용자에게는 GUI 형태의 사용자 인터페이스가 제공되며 사용자 인터페이스에 의해 작성된 질의는 시공간 이동 객체 관리 시스템에 의해 처리된다. GIS 기반 시공간 이동 객체 관리 시스템은 공간 관리 도구, 시공간 이동 객체 연산 처리기, 이동 정보 처리기와 같은 세 개의 서브 모듈로 구성된다.

공간 관리 도구는 공간 객체들을 관리하는 서브 모듈로써 공간 객체들을 저장 관리한다. 또한 이들 간의 관계 분석을 위한 공간 연산자를 제공하며, 공간 객체의 저장 및 관리를 위해 몇몇 데이터베이스 릴레이션을 필요로 한다. 시공간 이동 객체 연산 처리기는 시간에 따라 움직이는 시공간 이동 객체에 대한 기본적인 검색 연산과 시공간 이동 객체 관련 연산자들을 제공하는 서브 모듈이다. 데이터베이스 릴레이션을 검색하거나 이동 함수를 적용하여 주어진 질의를 처리할 수 있는 연산자를 적용한다. 이동 정보 처리기는 사용자가 입력한 이동 정보를 처리하는 서브 모듈로써 입력되는 이동 정보를 관리하며 데이터베이스 릴레이션을 갱신한다. 이때 행위 정보의 변경 여부를 검사하며 위치 정보의 이력을 유지한다.

4.2 데이터베이스 구성 및 연산

기존의 관계형이나 객체지향형 또는 객체관계형 데이터 모델들은 데이터가 명시적으로 갱신되지 않는 한 항상 일정한 값을 유지한다[3]고 가정하고 있다. 즉, 명시적 갱신 연산이 실행되기 전까지 데이터가 갖는 값은 항상 동일하다고 판단되는 것이다. 이와 같은 모델은 시공간 이동 객체를 저장 및 검색하기 위해 위치 정보를 지속적으로 갱신해야 하며, 시스템에 많은 부하를 주게되는 문제점이 발생된다. 따라서, 이 논문에서는 기존의 관계형 데이터베이스를 이용하면서도 빈번하게 데이터베이스를 갱신하지 않고, 시간 종속적인 함수에 의해 이동 객체의 위치 정보를 관리할 수 있도록 하였다.

앞의 4.1장에서 제시한 GIS 기반 시공간 이동 객체 관리 시스템을 위한 이동 객체 데이터베이스 구조를 설명하기 위해 이 논문에서 응용 시스템으로 개발한 전장분석 시스템의 이동 객체 데이터베이스의 예를 들어 설명한다.

데이터베이스 릴레이션은 공간 정보 릴레이션, 질의 릴레이션, 이동 정보 릴레이션으로 구성된다. 공간 정보 릴레이션은 공간 객체들을 저장하기 위해 유지되는 릴레이션이고, 질의 릴레이션은 현재 시공간 이동 객체들에 대해 질의를 수행하는 릴레이션으로 현재 시공간 이동 객체들의 행위 정보를 유지한다. 이동 정보 릴레이션은 입력된 이동 정보의 이력을 유지하는 릴레이션으로 행위 정보의 변경 여부를 검사하고 과거 정보 관련 질의를 수행하는데 사용된다.

<표 4> 질의 릴레이션

속성	ID	POSITION	ORIENT	SPEED	ORI_ERR	SPE_ERR	VTs	VTe
자료형	int	point	int	int	int	int	string	string
설명	키	시작 위치	이동 방향	이동 속도	방향 오차 범위	속도 오차 범위	시작 유효 시간	종료 유효 시간

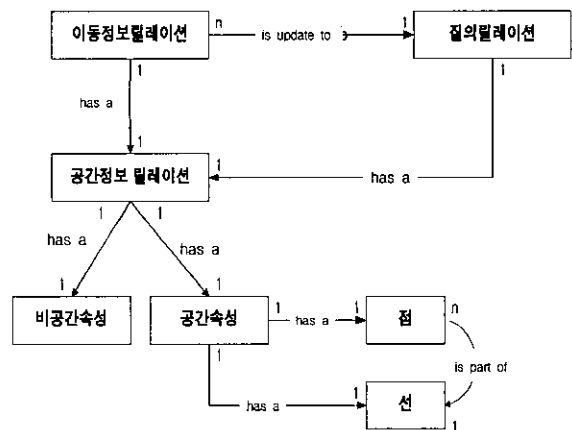
<표 4>의 질의 릴레이션은 각각의 시공간 이동 객체들에 대해 이동 위치에 관련된 질의를 수행하기 위해 유지된다. 이를 위해 행위 정보로 이동 방향과 이동 속도를 유지하며 이 행위 정보가 어느 시점부터 적용되는 지를 나타내는 시간 정보 및 어느 위치에서부터 계산되는 지를 나타내는 위치 정보를 포함한다.

<표 5> 이동 정보 릴레이션

속성	ID	ObjectID	PATH	ORIENT	SPEED	VTs	VTe
자료형	int	int	line	int	int	string	string
설명	키	객체 식별자	이동 경로	이동 방향	이동 속도	시작 유효 시간	종료 유효 시간

<표 5>의 이동 정보 릴레이션은 입력된 이동 정보들을 저장하는 릴레이션이다. 이동 정보는 행위 정보의 변경 여부를 검사한 후 행위 정보가 변경되지 않으면 이동 정보 릴레이션에 저장되고, 행위 정보가 변경된 경우 질의 릴레이션을 변경한 후 이동 정보 릴레이션에 저장된다. 이동 정보는 행위 정보의 변경 여부 검사 이외에도 과거 정보를 필요로 하는 처리에도 사용된다.

공간정보 릴레이션은 이동 객체의 위치를 나타내는 점이나 이동 경로를 나타내는 선 등의 공간 정보를 저장하기 위한 릴레이션이며 이동정보 릴레이션, 질의 릴레이션, 공간정보 릴레이션 간의 관계는 다음 (그림 10)과 같다.



(그림 10) 공간 정보를 표현한 개체관계 다이어그램

이동정보 릴레이션은 이동 정보를 입력받아 질의 릴레이션을 현재 이동 상태를 표현하도록 변경하며, 이동정보 릴레이션과 질의 릴레이션 모두 위치 표현을 위해 공간정보 릴레이션을 참조한다. 공간정보 릴레이션은 비공간 속성 표현을 위한 비공간 속성 릴레이션과 공간 정보 표현을 위한 공간속성 릴레이션을 가지며, 공간속성 릴레이션은 질의 릴레이션의 시작위치(<표 4>참조)를 표현하기 위한 점 릴레이션과 이동정보 테이블의 이동경로(<표 5>참조)를 표현하기 위한 선 릴레이션을 가진다. 선 릴레이션은 점들의 집합이다.

<표 6> 이동 객체 연산자

종 류	반 환 값
GetDistance	MReal
GetTrajectory	Line
GetLength	Real
GetMinValue, GetMaxValue	Int
GetMinTime, GetMaxTime	Time Point
GetSpeed	Int
PositionAtTime	Set of Location

<표 6>은 Erwig[4]가 제시한 시공간 이동 객체 연산자를 바탕으로 이 연구에서 사용될 시공간 이동 객체 연산자를 나타낸 것이다. 각각의 연산자는 기존의 공간 연산자나 산술 연산자에 시간 개념을 추가한 것이다. *GetDistance*는 모든 시간에 대해 이동 점들 간의 거리를 구한다. *GetDistance*에 의해 구해진 거리는 시간에 따라 값이 변하는 MReal[4]이 된다. *GetTrajectory*는 주어진 이동 점의 궤적을 구하여 선으로 표현한다. *GetLength*는 주어진 선의 총 길이를 구한다. *GetMinValue*와 *GetMaxValue*는 주어진 값들 중 최대 값이나 최소 값을 구하며, *GetMinTime*과 *GetMaxTime*은 최소, 최대 값을 가질 때의 시간을 구한다. *GetSpeed*는 주어진 시공간 이동 객체의 이동 속도를 구한다. *PositionAtTime*은 특정 시간에 시공간 이동 객체의 위치를 구하며, 이때 시간은 과거나 현재 또는 가까운 미래의 시간일 수 있다.

4.3 행위 정보 처리 알고리즘

행위 정보 처리란 시공간 이동 객체의 행위 정보인 속도와 방향을 처리하는 것이다. 속도 연산은 속도 평균에 의해 행위 정보의 변경 여부를 판단하는 방법을 사용하고 방향 연산은 고정 기준선을 사용하여 행위 정보의 변경 여부를 판단하는 방법을 사용한다. 이 두 가지 방법의 구체적인 행위 정보 변경 처리 알고리즘을 살펴보자.

알고리즘에서 사용되는 *DBmanager*는 JDBC를 사용해 구현한 클래스로 메소드별 전달자 및 기능은 다음 <표 7>에 나타나 있다.

<표 7> DBmanager의 메소드별 기능

메 소 드 명	기 능
selectQuery(String query)	질의를 수행하고 결과로 데이터셋을 생성
getValue(int row, int col)	데이터셋에서 row와 col에 의해 값을 구함
getBaseX(int ID)	기준점의 좌표
getBaseY(int ID)	
getBaseDate(int ID)	행위 정보가 설정된 시간을 검색
getBaseSpeed(int ID)	기준 속도
getBaseOrient(int ID)	기준 방향
getBaseSpError(int ID)	속도 오차의 크기
getBaseOrError(int ID)	방향 오차의 크기

다음 알고리즘 1은 속도 평균에 의해 행위 정보 변경을 처리하는 알고리즘이다.

```

public boolean IsSpeedChange(int ID, int x , int y, String date) {
    DBmanager dm = new DBmanager();
    String basetime = dm.getBaseDate(ID);
    int timesize = this.getDateDiff(date,basetime)

    /* Make query */
    query = " select * from HistoryTable " + "where date > "
           + basetime + " and ID = " + Integer.toString(ID)
           + " order by date";
    dm.selectQuery(query, historyInfo);

    /*Field order of result set is ID, X_Coord, Y_Coord, VTs, VTe*/
    int x, y, d, length;
    /* Add length of historical informations */
    for(int i=0;i<dm.getRecordsCount()-1;i++) {
        /* historical location of nth */
        x1 = dm.getValue(i,1);
        y1 = dm.getValue(i,2);
        /* historical location of n+1th */
        x2 = dm.getValue(i+1,1);
        y2 = dm.getValue(i+1,2);
        total = total + this.getLength(x1,y1,x2,y2);
    }

    /* Add length of current information */
    total = total + this.getLength(x2,y2,x,y);
    /* Compare current speed with base speed */
    int diff = abs(total/timesize - dm.getBaseSpeed(ID))
    if(diff > dm.getBaseSpError(ID))
        return true;
    else
        return false;
}
    
```

(알고리즘 1) 평균 속도에 의한 행위 정보 변경 알고리즘

알고리즘 1에서 볼 수 있는 것처럼 이 메소드의 기능은 행위 정보가 설정된 이후 현재까지 입력된 이동 정보들을 이동 정보 릴레이션에서 검색하여 총 이동 거리를 구하고 이를 이용하여 현재 평균 이동 속도를 계산하는 것이다. *basetime*은 행위 정보가 설정된 시간을 나타내며 *DBmanager()*로 전달될 질의문인 *query*는 이동 정보 릴레이션에서 *basetime*보다 이후에 입력된 이동 정보들을 시간순으로 검색하기 위한 SQL문이다. 이렇게 검색된 모든 이동 정보들에 대해 총 이동 거리를 계산하고 *basetime*을 이용하여 현

재 평균 이동 속도를 구한다. 이렇게 구해진 평균 이동 속도를 *DBmanager()*에서 전달받은 평균 이동 속도와 비교하여 주어진 오차의 범위를 넘을 경우 True를 반환한다. 이동 정보 입력 모듈은 반환값이 True인 경우 변경된 행위 정보를 수집하여 질의 릴레이션을 갱신한다.

고정 기준선에 의하여 행위 정보의 변경을 처리하는 알고리즘은 다음과 같다.

```
public boolean IsOrientChange(int ID, int x, int y) {
    DBmanager dm = new DBmanager();
    /* Base Location */
    int baseX      = dm.getBaseX(ID);
    int baseY      = dm.getBaseY(ID);

    /* Current Location */
    int currentX   = x;
    int currentY   = y;

    /* Get Length of current to base */
    int line1 = this.getLength(baseX, baseY, currentX, currentY);

    /* Get Length of base line */
    int line2 = this.getLength(baseX, baseY, currentX, baseY);

    /* Get each angles */
    int currentAlgle = this.GetAngle(line1, line2);
    int baseAlgle    = dm.getBaseAngle(ID);
    int diff         = abs(currentAlgle - baseAlgle)

    /* Compare */
    if(diff > dm.getBaseOrError(ID);)
        return True
    else
        return False
}
```

(알고리즘 2) 고정 기준선에 의한 행위 정보 변경 알고리즘

알고리즘 2의 메소드는 현재 위치와 기준점 및 기준선의 끝점 사이의 각을 구하고 이를 행위 정보에 입력된 방위 대표값 및 그 오차와 비교하여 이동 방향의 변경 여부를 결정한다. 이를 위해 기준점과 현재 위치 사이의 거리를 구하고 기준선의 거리를 구한다. 기준선의 오른쪽 끝점의 좌표는 기준점의 Y좌표와 현재 위치의 X좌표를 가지는 점이므로 이 점과 기준점과의 거리를 구하면 기준선의 거리를 구할 수 있다. 이 두 선의 길이를 이용하여 이동 방향을 구하고 변경 여부를 판단하여 이동 정보 입력 모듈로 해당되는 값을 전달한다. 이후의 이동 정보 입력 모듈의 행위는 속도 정보 변경 파악 알고리즘에서의 행위와 동일하다.

5 구 현

이 논문에서 제시한 시공간 이동 객체 관리 시스템의 적용 가능성을 보이기 위해 전장분석에 적용하여 구현한 결과를 예로서 설명하고, 실제 적용 가능성을 보인다.

5.1 환경

이 논문에서 구현한 시공간 이동 객체 관련 응용인 전장 분석 시스템은 클라이언트 시스템, 서버 시스템, 그리고 GIS 시스템으로 구성되었다. 클라이언트 시스템은 사용자에게 질의를 입력받아 이를 서버 시스템으로 전달한다. 서버 시스템은 전달받은 질의를 처리하기 위해 DBMS를 검색하고 GIS 시스템을 사용하여 결과를 클라이언트 시스템으로 전달한다.

서버 시스템은 UNIX 계열인 Solaris 운영체제를 사용하였고, 클라이언트는 Windows NT 4.0(package 4)을 사용하였다. 그리고, 하부의 데이터 관리를 위한 DBMS는 Oracle 7을 사용하였으며, 지도 관리를 위한 공간 관리 시스템은 Geowin 공간 관리 도구[17]를 사용하였다. Geowin은 순수 자바로 개발된 공간 관리 도구로 Java에서 제공되는 클래스들을 기초로 공간 객체 관리를 위한 클래스들을 제공하며 이 클래스들은 Java에서 제공되는 클래스들과 동일한 방법으로 사용 가능하다. 제안 시스템은 Java 프로그래밍 언어로 구현하였다.

5.2 결과 예시

실험 결과는 GIS 기반 시공간 이동 객체 관리 시스템의 응용인 전장 분석 시스템의 기능 중 시공간 이동 객체 관련 기능을 위주로 설명한다. 전장 분석 시스템에서 제공되는 기능은 여러 가지가 있지만, 이 논문에서 설명하는 기능은 시공간 이동 객체 관련 기능을 중심으로 이동 정보 입력, 일별 객체 위치 검색, 그리고 객체의 향후 이동 위치 검색에 관한 3가지 기능을 설명한다.

첫째, 다음 질의 1은 객체의 이동 정보를 입력하는 예이다. 사용자는 (그림 11)의 사용자 인터페이스를 통해서 원하는 부분에 내용을 입력하여 질의를 완성할 수 있다.

질의 1) "2000년 5월 6일 (345, 560) 위치에서 관측된 객체의 이동 정보를 입력하십시오."

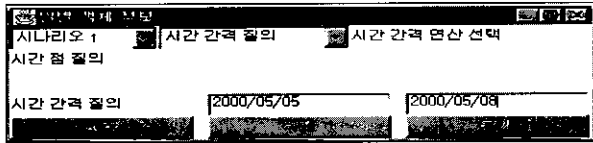
(그림 11) 이동 정보 입력

(그림 11)은 데이터베이스에 관측된 시공간 이동 객체의 이동 정보를 입력하는 화면이다. 이 화면은 2000년 5월 6일에 (345, 560) 지점에서 관측된 객체의 이동 정보를 입력하는 것을 보여준다. 입력된 이동 정보는 이동 정보 릴레이션에 삽입되며, 행위 정보의 변경여부 검사를 위한 처리가 동시에 진행된다.

둘째, 일별 객체 위치 검색은 사용자가 조회하고자 하는

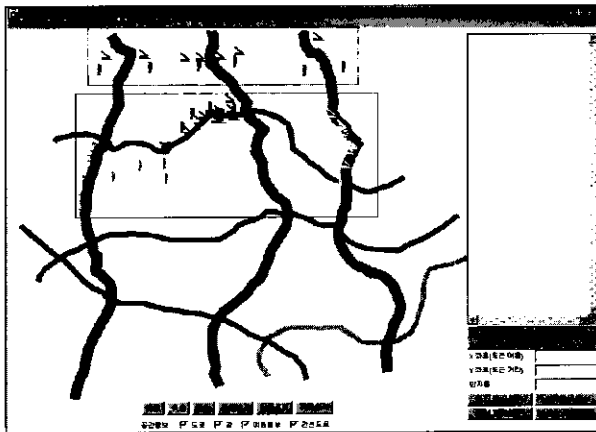
시나리오와 날짜를 입력하면 그 시점에 객체의 위치를 출력하는 기능이다. 사용되는 시간은 시간 점과 시간 간격 모두 가능하다. 다음 질의 2는 일별 객체 위치 검색을 위한 질의 예이다.

질의 2) “2000년 5월 5일부터 2000년 5월 8일까지의 모든 객체에 대해 객체 위치를 검색하라.”



(그림 12) 일별 객체 위치 검색 질의 작성

(그림 12)는 시나리오1에 대해 2000년 5월 5일부터 2000년 5월 8일까지의 객체의 위치 정보를 검색하는 질의 입력 화면이며 그 결과는 (그림 13)과 같다.



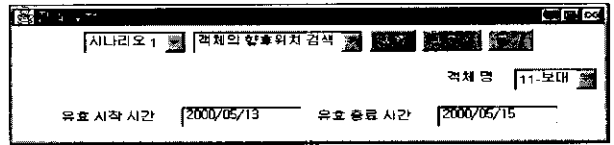
(그림 13) 일별 객체 위치 검색 결과

(그림 13)은 (그림 12)에서 작성된 질의에 대한 결과 화면이다. 질의 수행 결과 사용자가 볼 수 있는 정보는 두 가지이다. 첫째는, 해당 객체의 초기 위치 화면이다. 객체들의 초기 위치는 화면에서 상단 부에 점선으로 표시된 영역 내부이다. 초기 위치는 가장 처음에 관측된 위치 정보이며 이후부터 관측되는 위치 정보는 이동 정보가 된다. 둘째는, 질의에 입력된 날짜에 해당되는 객체들의 위치 정보이며 실선으로 그려진 사각형 박스로 표시된 부분이다. (그림 13)의 경우에는 2000년 5월 5일부터 2000년 5월 8일까지의 모든 객체의 위치를 출력하기 때문에 초기 객체의 수보다 더 많은 결과가 출력된다.

셋째, 시공간 이동 객체의 이동 위치 검색 기능은 하나의 객체가 향후 이동할 위치 좌표를 계산하여 그 결과를 화면에 출력하는 기능이다. 이동 위치 검색 기능에서 사용되는 시간은 시간 간격과 시간 점 모두 가능하지만 각각의 객체에 대해서만 적용가능하고 객체 그룹에는 적용할 수 없다. 다음의 질의 3은 객체의 이동 위치 검색 기능을 실행하기

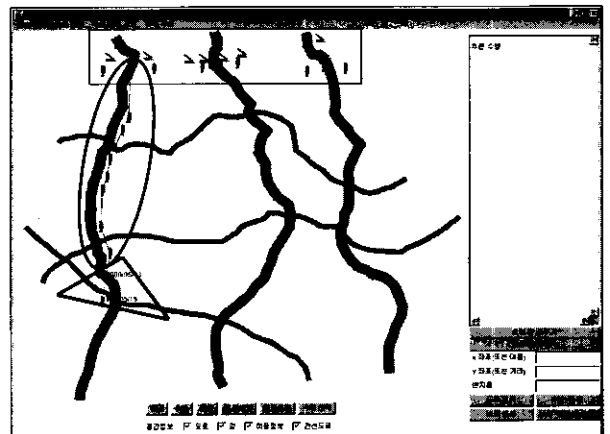
위한 질의 예이다.

질의 3) “객체 ‘11-보대’의 2000년 5월 13일부터 2000년 5월 15일까지의 이동 위치를 검색하라.”



(그림 14) 객체의 향후 이동 위치 검색 질의 작성

(그림 14)는 객체 ‘11-보대’에 대해 2000년 5월 13일부터 2000년 5월 15일까지의 객체의 이동 위치를 검색하는 질의 입력 화면이며 그 결과는 (그림 15)와 같다.



(그림 15) 객체의 향후 이동 위치 검색 결과

(그림 15)는 객체의 향후 이동 위치를 검색한 결과이다. 이 그림에서 사각형으로 묶여진 객체들은 이동 객체들의 초기 위치 정보를 나타내며 타원형으로 표시된 부분은 객체가 초기 위치에서부터 현재 위치까지 이동한 경로를 나타낸다. 그리고 삼각형으로 표시된 부분은 2000년 5월 13일부터 2000년 5월 15일까지의 객체가 이동할 예상 위치를 계산하여 표시한 결과이다.

6 결 론

기존의 데이터베이스 시스템을 이용하여 시공간 이동 객체를 관리할 경우 시간에 따라 이동 객체의 변경된 정보를 기록하기 위해 빈번한 갱신 연산이 발생되고, 갱신된 정보는 항상 현재 상태의 정보만을 유지하므로 과거 및 미래에 대한 정보를 전혀 제공하지 못하는 문제점이 발생된다. 이 논문에서는 이와 같은 문제점을 해결할 수 있는 시공간 이동 객체 관리 시스템을 제시하였다. 제안 시스템은 이동 객체의 정보를 위치 정보와 행위 정보로 구분하여 표현하며, 향후 객체의 이동 위치 검색을 위한 시간 종속적 함수를

제공하고, 행위 정보 변경 처리 알고리즘을 이용한 행위 정보의 유지 및 관리가 가능하다. 특히, 발생하는 모든 이력 정보의 저장 없이도 객체의 과거, 현재, 그리고 가까운 미래에 대한 위치 정보의 제공이 가능한 특징을 가진다.

제안 시스템의 적용가능성을 보이기 위해 이동 객체의 응용분야인 전장분석에 적용하여 시공간 이동 객체 관리 시스템을 구현하였다. 구현된 시스템은 관계형 데이터베이스를 확장하여 시공간 데이터베이스를 구축하고 이동 객체들의 위치 정보를 표현하였다. 또한, 시공간 연산의 효율적 수행을 위해 상용 GIS 도구를 활용하고, 시공간 이동 객체들의 위치 정보 관리를 위해 행위 정보를 유지하여 이동 객체의 위치 정보에 관한 질의 응답에는 행위 정보를 매개 변수로 하는 이동 함수를 적용하였다. 구현 시스템의 수행 결과를 예를 들어 살펴보았으며, 이를 통해 관계형 데이터베이스 시스템과 GIS 도구를 이용하여 실세계의 시공간 이동 객체를 효과적으로 관리할 수 있음을 확인하였다.

이 연구에서는 이동 함수의 적용에 있어 불확실성에 대한 처리를 수행하지 않았기 때문에 이동 함수의 계산 결과에 오차가 발생하는 문제를 가지며 행위 정보의 수정이 오랜 기간 동안 수행되지 않는다면 이 오차의 범위가 증가될 수 있다. 따라서, 향후 연구에서는 이러한 오차를 최소화할 수 있는 불확실성 처리에 대한 연구가 수행될 것이다.

참 고 문 헌

[1] 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," Fern Universitat Hagen, Informatik-Report pp.238, Sep. 1998. "http://www.informatik.fernuni-hagen.de/import/pi4/gueting/bibliography.html"

[2] 이기준, 김경숙, "이동객체의 움직임에 대한 시간적 명세", 한국정보과학회, 데이터베이스연구회지, 제16권, 1호, 2000년 8월.

[3] O. Wolfson, "Moving Object Databases : Issues and Solutions," In Proc. Of the 10th Intl. Conference on Scientific and Statistical Database Management, Capri, Italy, 1998.

[4] 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, Dec. 1997.

[5] M. P. Armstrong, "Temporality in Spatial Databases", Proceedings : GIS/LIS, 1988.

[6] G. Langran, "A Framework for Temporal Geographic Information Systems," Cartographica, 25(3), 1988.

[7] Donna J. Peuquet and Miu Duan, "An event-based spatio-temporal data model(ESTDM) for temporal analysis of geographic data," International Journal of Geographic Information Systems, 9(1), 1995.

[8] A. Montani and B. Pernki, "Temporal reasoning," In Temporal Databases : Theory, Design and Implementation, The Benjamin/Cummings Publishing Company Inc., 1993.

[9] A. Renolcn, "The Spatio-temporal Object Model : A Conceptual Model Based on Ontology," article submitted to Geoinformatika. 1997.

[10] A. Segev and A. Shoshani, "A Temporal Data Model Based on Time Sequences," In Temporal Database : Theory, Design, and Implementation, The Benjamin/Cummings Publishing Company Inc., 1993.

[11] M. Yuan, "Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatio-Temporal Queries," In Press by Transactions in GIS, 1997.

[12] M. F. Worboys, H. M. Hearchshow and D. J. Maguire, "Object-Oriented Modelling for Spatial Databases," Int. journal of GIS, 4(4), 1990.

[13] 김동호, "시공간 데이터베이스 질의처리 시스템," 충북대학교 전자계산학과, 박사학위논문, 1999년 2월.

[14] J. F. Allen, "Maintaining Knowledge about temporal intervals," CACM(Communication of the ACM), 26(11), pp.832-843, 1983.

[15] Luca Forlizzi, Ralf Hartmut Guting, Enrico Nardelli and Markus Schneider, "A Data Model and Data Structures for Moving Objects Databases," ACM SIGMOD Conference, pp.319-330, 2000.

[16] A. Prasad Sistla, Ouri Wolfson, Sam Chamberlain, Son Dao, "Modeling and Querying Moving Objects," ICDE(International Conference on Data Engineering), pp.422-432, 1997.

[17] Geostchnology, "Geowin System", "http://www.geowin.net"



신 기 수

e-mail : ksshin@kt.co.kr
 1973년 한국항공대학교 전자공학과(공학사)
 1986년 한양대학교 산업대학원 전자통신과(공학석사)
 1999년 충북대학교 대학원 전자계산학과(박사수료)

1978년 KTRI(전 한국전자통신연구원) 입사
 1984년~현재 한국통신 연구개발본부 가입자망연구소 근무
 관심분야 : 이동객체 및 이동통신, 홈네트워크, 이동서비스 등



안 윤 애

e-mail : yeahn@dblab.chungbuk.ac.kr
 1993년 한남대학교 전자계산공학과(학사)
 1996년 충북대학교 대학원 전자계산학과(이학석사)

1999년~현재 충북대학교 대학원 전자계산학과 박사과정
 관심분야 : 이동 객체 데이터베이스, 시공간 데이터베이스, 지리정보 시스템, 정보검색 시스템



배 증 철

e-mail : jcbae@dblab.chungbuk.ac.kr
1999년 군산대학교 전자계산학과(학사)
1999년~현재 충북대학교 대학원 전자계산학과 석사과정
2001년~현재 (주)우린정보 기술연구소
관심분야 : 시공간 데이터베이스, 정보검색 시스템, 지리정보시스템



정 영 진

e-mail : yjjeong@dblab.chungbuk.ac.kr
2000년 충북대학교 전자계산학과(학사)
2000년~현재 충북대학교 대학원 전자계산학과 석사과정
관심분야 : 지리정보 시스템, 이동 객체 데이터베이스



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산학과(이학사)
1980년 연세대학교 산업대학원 전산전공(공학석사)
1988년 연세대학교 대학원 전산전공(공학박사)
1976년~1986년 육군군수 지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송대 전산학과(조교수) 근무
1989년~1991년 Univ. of Arizona Research Staff(TempIS 연구원, Temporal DB)
1986년~현재 충북대학교 컴퓨터과학과 교수
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, GIS 및 지식기반 정보검색 시스템, 데이터마이닝, 객체와 지식베이스 시스템