

다차원 지리정보시스템을 위한 저장기법 및 분리된 저장구조

박 동 선[†] · 배 해 영^{††}

요 약

최근에 기존의 2차원적인 지리정보시스템에 시간차원을 포함하는 다차원 지리정보 데이터를 저장하기 위한 연구가 진행되고 있지만 대부분이 실제 구현에는 적용하기 어려운 개념적인 데이터모델의 설계단계에 머물고 있다. 본 논문에서는 기존 모델들의 문제점인 데이터 중복을 최소화 하고, 시간을 포함하는 질의에 대한 연산을 빠르게 수행할 수 있는 저장 기법인 MBNF기법을 설계하고, 데이터의 시간적 특성을 고려해서 과거 데이터베이스와 현재 데이터베이스에 분리하여 저장하는 분리된 저장구조를 설계하며, 분리된 데이터베이스간의 데이터 이동 알고리즘과 시간을 포함하는 질의를 효율적으로 수행하기 위한 검색 알고리즘을 설계한다.

Storage Strategy and Separated Storage Structure for Multi-dimensional Geographic Information System

Dong-Seon Park[†] · Hae-Young Bae^{††}

ABSTRACT

Up to the present, many GIS logical data models have been proposed to incorporate time dimension into traditional 2-dimensional (2D space) GIS databases. However, these models are difficult to implementation of multi-dimensional GIS and have problems of significant data redundancy and search performance. This paper proposes a new storage strategy and a separate storage structure, that is composed of current database and past database, in order to reduce data redundancy and improve search performance of temporal query. Also we design moving algorithm for migration from current database to past database, and design efficient search algorithm for temporal query.

1. 서 론

기존의 지리정보시스템들은 2차원 공간을 이용하여 공간객체의 위치를 표현하고 있으며, 공간객체의 정보가 변경되는 경우 이를 시스템에 반영하게 되면 갱신 연산 수행에 의하여 갱신 전의 정보들은 삭제되는 스

냅샷 시스템이다[1].

최근 환경영향평가, 도시계획, 재난방지 등의 분야에서 지리정보시스템을 이용하고자 하는 노력이 진행되고 있으며, 이들 분야에서는 현재시점과 과거시점의 공간객체 상태를 표현하는 정보에 대한 관리를 요구하고 있다. 이러한 요구를 수용하기 위하여 90년대 초반부터 시간을 포함하는 다차원 지리정보시스템에 대한 연구가 진행되고 있다. 이러한 연구들은 대부분 기존의 관계형 데이터베이스 시스템을 기반으로 하는 개념적인 시간 지원 지리정보 데이터모델에 대한 연구[2, 3,

* 본 연구는 정보통신부의 대학 S/W연구센터 지원사업에 의하여 수행되었음.

† 중 회 원 : 인하대학교 대학원 전자계산공학과

†† 종 신 회 원 : 인하대학교 전자계산공학과 교수

논문접수 : 1999년 11월 2일, 심사완료 : 1999년 12월 24일

1.5.6)와 이들 모델을 이용한 연산 및 질의어에 대한 연구[7,8]에 치중하고 있으며, 실제 시스템 구현에 필요한 데이터 저장 방법에 대한 연구는 많지 않다. 또한 기존에 연구된 개념 데이터모델들은 심각한 데이터 중복을 발생시킬 수 있으며, 시간을 포함하는 질의어에 대한 응답 시간이 효율적이지 못하다[9,10].

본 논문에서는 과거데이터에 대한 중복을 최소로 하여 전체 데이터베이스의 크기를 최소로 할 수 있는 저장기법과 시간을 포함하는 질의어에 대한 빠른 응답을 할 수 있는 분리된 지리정보 저장구조를 제안하며, 제안하는 저장기법과 저장구조를 위한 데이터 이동 알고리즘과 시간을 포함하는 질의어에 적용할 수 있는 검색 알고리즘을 설계한다. 시간을 포함하는 공간객체의 정보는 공간객체의 유효한 시간을 기준으로 과거데이터와 현재데이터로 분류할 수 있으며, 이들은 각각 과거 데이터베이스와 현재 데이터베이스에 저장하도록 한다. 분리된 저장구조에서는 시간을 포함하는 질의를 효과적으로 수행할 수 있으며, 특히 현재시점에 대한 질의는 현재 데이터베이스만을 대상으로 질의가 수행되기 때문에 기존의 스냅샷 시스템과 동일한 질의 수행 성능을 갖는다. 또한 과거 데이터베이스에는 제안하는 저장기법을 이용하여 중복이 최소화된 데이터를 저장할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 시간 지원 데이터모델에 대한 기존의 연구와 저장구조 설계에 필요한 저장 기법에 대하여 살펴본다. 3장에서는 본 논문에서 제안하는 분리된 저장구조와 데이터베이스의 내부 저장구조를 설계하며, 데이터베이스간의 데이터 이동 알고리즘과 검색 알고리즘을 설계한다. 4장에서는 모의 데이터를 이용하여 기존의 방법과 본 논문에서 제안하는 저장구조의 성능을 비교하며, 마지막 5장에서는 결론과 함께 본 논문에서 제안하는 저장구조와 관련된 향후 연구 과제에 대하여 살펴본다.

2. 관련 연구

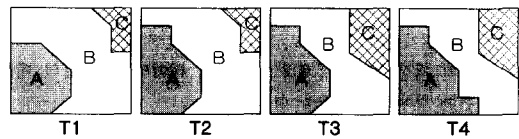
2.1 기존의 개념 지리정보 데이터 모델

Yuan[11]은 지리정보시스템의 시간 지원 모델을 레이어 단위, 어트리뷰트 단위, 공간객체 단위 시간 지원 모델로 구분하였으며, 이들은 각각 관계형 데이터베이스 시간 지원 모델인 릴레이션 단위, 튜플 단위, 어트리뷰트 단위 시간 지원 모델과 유사한 특징을

갖는다[12,13].

2.1.1 레이어 단위 시간 지원 데이터 모델

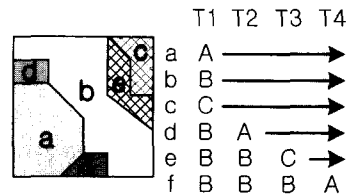
스냅샷 모델[14]이 대표적이며, 공간 정보의 변경이 발생할 때마다 변경된 공간객체가 포함된 레이어를 구성하는 모든 공간객체에 대한 새로운 스냅샷 레이어를 생성한다. 이 모델은 현재의 지리정보시스템을 이용하여 쉽게 구현할 수 있지만, 정보의 변화가 없는 공간객체의 데이터들도 새로 생성되는 레이어에 포함되기 때문에 매우 심각한 자료의 중복이 발생한다[11].



(그림 1) 스냅샷 모델의 예

2.1.2 어트리뷰트 단위 시간 지원 데이터 모델

시공 복합 모델(Space-time composite model)[15] 대표적이며, 하나의 레이어에 변화된 모든 공간객체들의 형상을 표현한다. 공간객체의 변화된 부분들에 대하여 새로운 식별자를 부여하기 때문에 공통된 속성을 가지는 하나의 공간객체가 여러 개의 분할된 객체로 표현된다. 즉, 동일한 객체가 여러 개의 튜플에 분산되어 저장되기 때문에, 객체의 변경된 회수만큼 데이터의 중복이 발생하게되며, 어느 시점에서 공간객체의 상태를 검색하기 위해서는 여러 튜플의 조인이 필요하다. 또한, 단 하나의 공간객체가 변경된다라도 데이터베이스의 모든 내용을 재구성해야 한다[11].

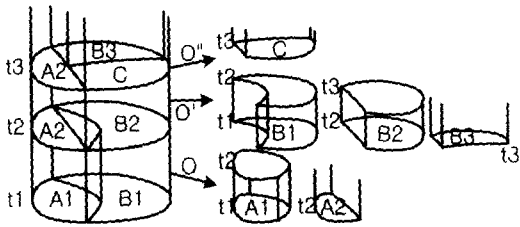


(그림 2) 시공 복합 모델의 예

2.1.3 공간객체 단위 시간 지원 데이터 모델

시공간 객체 모델(spatiotemporal object model)이 [16] 대표적이며, 2차원 공간에 시간 차원이 결합된 3차원 공간으로 공간객체를 표현한다. 각 공간객체들은

시간과 공간적으로 동일한 특성을 갖는 시공간 원자 (A1, A2, B1, B2, B3, C)로 구성되며, 시공간 객체(O, O', O'')는 시간과 공간에 대한 변화 과정을 표현한다. 이 모델에서는 시간의 흐름에 따른 객체의 변화 과정을 쉽게 기록할 수 있지만, 전체적인 공간의 변화 과정은 표현할 수 없다.



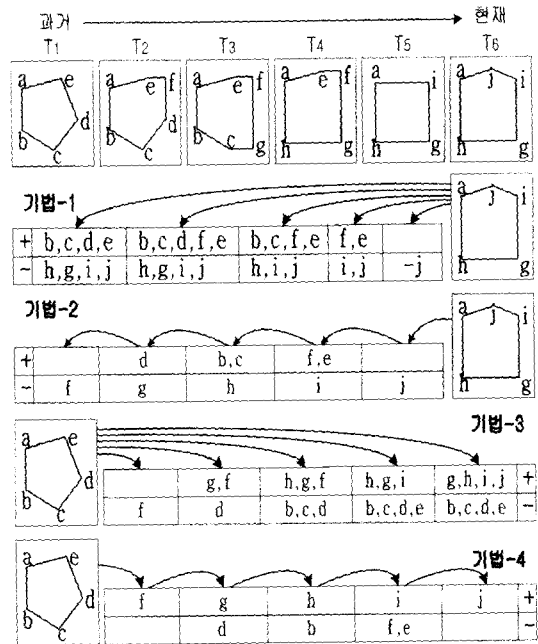
(그림 3) 시공간 객체 모델의 예

2.2 기존의 이력 정보 저장 기법

Dadam[9]과 Langran[10]은 객체의 변화에 따른 데이터 저장기법을 다음과 같이 분류하였다. $P_o(T_n)$ 은 시점 T_n 에서 객체 o 와 연관된 모든 공간 정보를 저장함을 의미하며, $\Delta_o(T_{n-1}, T_n)$ 은 객체 o 의 두 시점 T_{n-1} 과 T_n 일 때의 정보를 비교하여 변화(추가 또는 삭제)된 정보만 저장함을 의미한다. T_1 은 최초 시점이며, T_n 은 현재시점이다. (그림-4)는 하나의 객체에 대하여 다음의 기법들을 적용한 예이다.

- 기법-1 : backward oriented amendments to a base state
 $\Delta_o(T_1, T_n), \Delta_o(T_2, T_n), \dots, \Delta_o(T_{n-1}, T_n), P_o(T_n)$
- 기법-2 : backward oriented cumulative
 $\Delta_o(T_1, T_2), \Delta_o(T_2, T_3), \dots, \Delta_o(T_{n-1}, T_n), P_o(T_n)$
- 기법-3 : forward oriented amendments to the base state
 $P_o(T_1), \Delta_o(T_1, T_2), \Delta_o(T_1, T_3), \dots, \Delta_o(T_1, T_{n-1}), \Delta_o(T_1, T_n)$
- 기법-4 : forward oriented cumulative amendments
 $P_o(T_1), \Delta_o(T_1, T_2), \Delta_o(T_2, T_3), \dots, \Delta_o(T_{n-2}, T_{n-1}), \Delta_o(T_{n-1}, T_n)$

기법-1 : 현재시점의 상태를 기준상태(base state)로 하며, 과거시점의 상태들은 기준상태와 비교하여 변화된 정보들만 저장한다. 현재시점에 대한 검색 연산은 매우 빠르게 수행될 수 있으며, 과거시점의 검색도 현재상태와 한번만 비교하면 되기 때문에 비교적 빠르게 수행될 수



(그림 4) 단일 공간객체를 대상으로 하는 이력정보 저장 기법의 예

있다. 그러나 현재시점이 변경될 때마다 모든 과거데이터를 재구성해야 한다. 또한, 기준상태로부터 공간 정보의 변경 회수가 많아질 경우 변경된 객체의 상태를 표현하기 위하여 객체의 원형 데이터(original data)보다 더 많은 데이터를 저장해야 하며, 데이터들이 중복되어 표현된다.

기법-2 : 현재시점의 상태는 모든 데이터를 저장하고, 과거시점의 상태들은 바로 다음 시점의 상태와 비교하여 변화된 데이터들만 저장한다. 데이터 중복은 해결하였지만, 과거시점에 대한 검색은 현재시점의 상태로부터 시작하여 반복되는 비교 연산을 수행해야 하기 때문에 현재시점으로부터 거리가 멀어질수록 검색 시간이 점점 오래 걸리게 된다.

기법-3 : T_1 시점의 상태를 기준상태로 하며, 이후 시점의 상태들은 기준상태와 비교하여 변화된 데이터만 저장한다. 현재와 과거시점의 상태에 대한 검색은 기준상태와 한번의 비교 연산으로 수행되기 때문에 과거정보의 누적에 영향을 받지 않고 빠르게 수행될 수 있다. 그러나 기준

상태로부터 공간 정보의 변경 회수가 많아질 경우 변화된 객체의 상태를 표현하기 위하여 객체의 원형 데이터보다 더 많은 데이터를 저장해야 하며, 동일한 데이터들이 중복된다.

기법-4: T_1 시점의 상태는 모든 데이터를 저장하며 이후 시점의 상태들은 바로 전 시점의 상태와 비교하여 변화된 데이터들만 저장한다. 데이터 중복은 해결하였지만 인접된 시점간에 변화된 데이터가 많은 경우에는 원형 데이터보다 더 많은 데이터를 저장해야한다. 또한 T_1 시점을 제외한 다른 시점들에 대한 검색이나 저장을 위하여 바로 전 시점의 완전한 상태를 알아야 하며 이를 위하여 T_1 시점부터 반복되는 비교 연산을 수행해야 하기 때문에 과거정보가 누적되어감에 따라 연산시간이 점점 느려지게 된다.

3. 분리된 지리정보 저장구조 및 연산 알고리즘 설계

본 논문에서는 데이터의 검색 및 저장 효율을 위하여 객체 정보의 유효시간을 기준으로 과거 데이터베이스와 현재 데이터베이스로 분리된 저장구조를 설계한다. 또한, 저장되는 데이터의 중복을 최소로 할 수 있는 저장기법과, 분리된 저장구조간의 데이터 이동 알고리즘 및 검색 알고리즘을 설계한다. 본 논문에서 설계하는 저장구조에서는 공간 객체 단위로 시간을 지원한다.

3.1 분리된 저장구조 설계

본 논문에서 설계하는 분리된 저장구조는 현재시점에 유효한 정보는 현재 데이터베이스에 저장하며, 과거시점에 유효한 데이터는 과거 데이터베이스에 저장한다. 현재 데이터베이스는 공간객체들의 정보를 저장하는 현재데이터블로 구성되며, 과거 데이터베이스는 공간객체의 정보를 저장하는 과거데이터블과, 각 공간객체의 기준상태 정보를 저장하는 기준상태 메타데이터블로 구성된다.

3.1.1 현재 데이터베이스

현재 데이터베이스를 구성하는 현재데이터블은 (그림 5)와 같다. '객체ID'는 공간객체 식별자이며 현재데이터블의 주요키(primary key)로 사용한다. 'Node'는 공간객

체를 구성하는 모든 노드를 저장하며, 'VS'와 'TS'는 각각 유효시작시간과 거래시작시간이다. 유효시작시간은 사용자가 입력한 시간이 저장되며, 거래시작시간은 시스템에서 자동으로 부여한다. 현재데이터블에서 관리하는 객체들의 유효종료시간과 거래종료시간은 항상 현재시점이기 때문에 이들에 대한 정보는 저장하지 않는다.

객체ID	Node	VS	TS
------	------	----	----

(그림 5) 현재데이터블

3.1.2 과거 데이터베이스

과거 데이터베이스에는 현재 데이터베이스로부터 공간정보의 변경이 발생한 공간객체의 정보를 관리하며, 객체의 공간정보는 3.2에서 제안하는 MBNF기법을 이용하여 기준상태로부터 변화된 노드나, 객체를 구성하는 모든 노드를 저장하여 과거 데이터베이스에 저장되는 데이터의 양을 최소로 한다.

과거 데이터베이스는 과거데이터블과 기준상태 메타데이터블로 구성된다. 과거데이터블에 저장되는 공간정보는 기준상태일 경우는 공간정보를 구성하는 모든 노드를 'UNode'에 저장하며, 기준상태가 아닌 경우는 기준상태와 비교하여 추가되는 노드는 'UNode'에 저장하고, 삭제되는 노드는 'DNode'에 저장한다. 또한 유효시간과 거래시간의 종료시간은 각각 'VE'와 'TE'에 저장한다. 과거데이터블에는 동일한 객체에 대하여 여러 개의 튜플이 존재할 수 있기 때문에 '객체ID'와 'VS'(또는 'TS')를 이용한 복합키를 주요키로 한다.

기준상태 메타데이터블은 객체식별자와, 기준상태가 적용될 유효시간과 거래시간의 범위를 포함하며, '객체ID'와 'VS'(또는 'TS')를 이용한 복합키가 주요키가 된다. 기준상태에 대한 공간정보는 과거데이터블에 저장되기 때문에 기준상태 메타데이터블에는 저장하지 않는다.

객체ID	UNode	DNode	VS	VE	TS	TE
------	-------	-------	----	----	----	----

(a) 과거데이터블

객체ID	VS	VE	TS	TE
------	----	----	----	----

(b) 기준상태 메타데이터블

(그림 6) 과거데이터블과 기준상태 메타데이터블

3.2 복수의 기준상태를 갖는 전진개정기법

본 절에서는 2장에서 살펴본 기법-3을 개선하여 현재 및 과거시점의 상태에 대한 검색 연산 시간이 과거 데이터의 누적에 영향을 받지 않으며, 과거 데이터베이스의 크기를 최소로 할 수 있는 복수의 기준상태를 갖는 전진개정기법(MBNF: Multi Base state aNd Forward-oriented amendment method)을 제안한다.

제안하는 MBNF기법은 다음과 같이 표현할 수 있다.

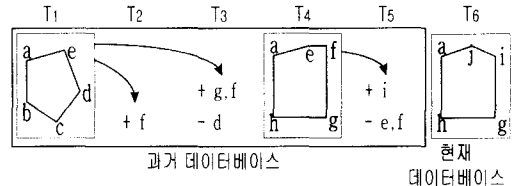
$$P_0(T_1), \Delta_0(T_1, T_2), \Delta_0(T_1, T_3), \dots, P_0(T_m), \Delta_0(T_m, T_{m+1}), \dots, \Delta_0(T_m, T_{n-1}), \Delta_0(T_1, T_n)$$

MBNF기법을 이용한 현재 및 과거 데이터베이스에 대한 데이터 저장 방법은 다음과 같다. 현재시점에 유효한 데이터는 레이어를 구성하는 모든 공간객체의 상태를 현재 데이터베이스에서 관리하며, 과거 데이터베이스에는 레이어를 구성하는 객체들 중 변경된 객체의 데이터만 관리한다. 어느 객체 O_i 가 T_n 시점에서 최초로 갱신이 발생하였을 경우 T_n 시점에서 객체 O_i 를 구성하는 모든 공간 데이터를 현재 데이터베이스에 저장하며, T_{n-1} 시점의 데이터는 과거데이터가 되어 현재 데이터베이스로부터 과거 데이터베이스로 이동한다. 객체 O_i 에 대한 갱신이 처음 발생하였기 때문에 T_{n-1} 시점의 데이터는 객체 O_i 에 대한 최초의 기준상태가 되어 객체를 구성하는 모든 공간 데이터를 과거 데이터베이스에 저장하며, 기준상태를 식별할 수 있는 최소의 정보를 기준상태 메타데이터에 저장한다. T_{n-m} ($m \geq 1$)시점에서 객체 O_i 에 대한 갱신이 발생하면 T_{n-m-1} 시점에 현재 데이터베이스에 저장되어있던 데이터를 과거 데이터베이스로 이동하며, 객체 O_i 의 기준상태 데이터와 비교하여 변화된 공간 데이터를 저장한다. 이때 변화된 공간 데이터의 양이 T_{n-m-1} 시점에 현재 데이터베이스에 저장되어있던 데이터의 양보다 많을 경우에는 T_{n-m-1} 시점에서 객체 O_i 를 구성하는 모든 공간 데이터를 과거 데이터베이스에 저장하며, 이 데이터를 객체 O_i 에 대한 새로운 기준상태로 한다.

MBNF기법은 현재시점에 유효한 데이터는 현재 데이터베이스에 완전한 상태로 관리하기 때문에 현재시점에 유효한 데이터에 대한 질의 처리는 누적되는 과거데이터에 영향을 받지 않고 매우 빠르게 수행할 수 있으며, 과거시점에 대한 질의 처리도 기준상태와 한번만 비교 연산을 수행하면 된다. 또한, 과거 데이터베이스에는 복수의 기준상태를 이용하여 기존의 방법보

다 적은 양의 데이터를 저장할 수 있으며, 중복되는 데이터의 양을 줄일 수 있다.

(그림 7)은 (그림 4)의 예를 MBNF기법에 적용한 결과이다. 시점 T_2 시점에서는 T_1 시점의 상태가 과거 데이터베이스로 이동하며 최초의 기준상태가 된다. T_3 과 T_4 시점에서는 각각 T_2 와 T_3 시점의 상태가 과거 데이터베이스로 이동하며 이들은 기준상태와 비교하여 변화된 노드의 수가 원형 상태의 노드 수보다 적기 때문에 변화된 노드를 저장한다. T_5 시점에서는 T_4 시점의 상태가 과거 데이터베이스로 이동하며 기준상태와 비교한 결과의 노드 수가 원형 상태의 노드 수보다 많기 때문에 객체의 원형 상태를 모두 저장함과 동시에 새로운 기준상태가 된다. 현재 데이터베이스에는 현재 시점인 T_6 시점의 상태를 모두 저장한다.



(그림 7) MBNF기법을 이용한 데이터의 저장 예

3.3 데이터 이동 및 검색 알고리즘

본 절에서는 현재 데이터베이스와 과거 데이터베이스간의 데이터 이동 알고리즘과, 시간이 포함된 검색 알고리즘을 설계한다. 본 절에서 설계하는 알고리즘에서 언급하는 시간은 유효시간과 거래시간을 특별히 구분하지 않는다.

알고리즘에 사용되는 변수의 의미와 함수의 기능은 각각 다음과 같으며, 각 함수에 사용된 알고리즘들은 튜플을 구성하는 어트리뷰트 값의 비교와 테이블로부터 튜플의 삭제 및 추가 또는 키값을 이용한 검색연산으로 구성되기 때문에 함수에 대한 알고리즘의 기술은 생략한다.

- Onew : 객체 O의 갱신된 정보.
- Oh : 현재테이블에서 과거테이블로 이동할 객체 O의 데이터.
- Op : 과거테이블의 객체 O의 정보.
- Bi : 과거테이블로부터 검색한 객체의 기준상태
- BSo : 객체 O에 대한 기준상태 데이터.
- VS, VE : 유효시작시간과 유효종료시간.

- TS, TE : 거래시작시간과 거래종료시간.
- now : 현재시점을 의미하는 시간 변수.
- Key : 객체 ID + 유효 시작시간.
- AddToTbl(T,O) : 객체 O를 테이블 T에 저장.
- CreateBaseState(O) : 객체 O의 기준상태 정보 생성.
- CreateChangeData(Object, BaseState) : 기준상태의 데이터와 과거데이터로 이동할 데이터를 비교하여 변화된 노드를 생성.
- DeleteFromCur(O_ID) : 현재테이블로부터 객체의 정보를 삭제, 삭제된 객체의 정보를 반환.
- FindBaseTbl(O_ID,Time) : 기준상태 메타데이터로부터 객체의 기준상태를 검색.
- ReadPastTbl(Time) : 과거테이블에서 Time의 시점에 유효한 객체의 정보를 튜플단위로 검색.
- OutputResult(O) : 객체의 정보를 출력.
- AddNode(O-1, O-2) : 객체-1의 노드에 객체-2의 UNode 내용을 추가.
- DelNode(O-1, O-2) : 객체-1의 노드로부터 객체-2의 DNode 내용을 제거.
- FindPastTbl(Key) : Key를 이용하여 과거테이블로부터 튜플을 검색.
- OutputCurTbl(Time) : 현재테이블로부터 Time의 시점에 유효한 모든 객체의 정보를 출력.

3.3.1 데이터 이동 알고리즘

공간정보에 대한 갱신이 발생하였을 경우 현재 데이터베이스에 저장되어 있던 정보는 유효종료시간과 거래종료시간이 추가되어 과거 데이터베이스로 이동하며, 이 때 기준상태와 비교하여 변화된 데이터를 추출하여 과거테이블에 저장될 정보를 생성한다. 생성된 정보는 데이터의 중복 저장 및 객체의 원형 데이터보다 더 많은 데이터를 저장하는 경우를 방지하기 위하여 원형 데이터와 비교 과정을 수행한 후 과거테이블에 저장할 정보를 결정한다. 이동 알고리즘은 (그림 8)과 같으며, 문장 앞의 원문자는 4장의 성능평가부분에서 데이터 이동 알고리즘의 분석을 위하여 표시하였다.

3.3.2 시간이 포함된 검색 알고리즘

검색시간이 현재시점일 경우는 현재테이블에 저장되어 있는 모든 정보들을 출력하면 된다. 검색시간이 과거시점일 경우는 현재 데이터베이스와 과거 데이터베이스에 저장되어있는 정보들이 연산 대상이 된다. 현

```

Migration(Onew.ID)
{
    Oh = DeleteFromCur(Onew.ID);
    ① Oh.VE = Onew.VS;
    ② Oh.TE = Onew.TS;
    AddToTbl(CurrentTable, Onew);
    BSo = FindBaseTbl(Oh.ID,Oh.VS);
    if(BSo == NULL) {
        AddToTbl(PastTable,Oh);
        BSo = CreateBaseState(Oh);
        AddToTbl(BaseStateTable,BSo);
    } else {
        Op = CreateChangeData(Oh,BSo);
        ③ if(Op의 노드 수 > Oh의 노드 수) {
            ④ AddToTbl(PastTable,Oh);
            ⑤ BSo = CreateBaseState(Oh);
            AddToTbl(BaseStateTable,BSo);
        } else AddToTbl(PastTable,Op);
    }
}
    
```

(그림 8) 데이터 이동 알고리즘

재 데이터베이스에서는 현재테이블의 유효시간을 참조하여 검색시간을 포함하는 객체의 정보들을 검색 결과로 출력하며, 과거 데이터베이스에서는 과거테이블과 기준상태 메타데이터를 이용하여 검색시간을 포함하는 객체의 정보를 검색한다. (그림 9)의 알고리즘에서 공간 조건에 의한 검색 과정은 생략하였으며, 그 이유는 4장의 검색시간 분석 부분에서 기술한다.

```

Retrieve(time)
{
    if(time == now) {
        OutputCurTbl(now);
    } else {
        OutputCurTbl(time);
        while(! eof (Op = ReadPastTbl(time))) {
            BSo = FindBaseTbl(Op.ID,time);
            if(Op.Key == BSo.Key) {
                OutputResult(Op);
            } else {
                Bi = FindPastTbl(BSo.Key);
                AddNode(Bi,Op);
                DelNode(Bi,Op);
                OutputResult(Bi);
            } /* end of if */
        } /* end of while */
    }
}
    
```

(그림 9) 데이터 검색 알고리즘

4. 모의 데이터를 이용한 성능 평가

MBNF기법과 분리된 저장구조를 이용하여 구축한 데이터베이스와 2장에서 소개한 기존의 기법 중에서 비교적 데이터의 중복이 심하지 않고 과거 및 현재시점의 데이터에 대한 검색 연산시간이 효율적인 기법-3을 이용하여 구축한 데이터베이스를 분석 대상으로 하여 시점의 변화에 따른 데이터베이스의 크기와 연산 시간을 측정하여 성능을 평가한다.

4.1 성능평가 환경 및 방법

성능평가는 1,252개의 폴리곤으로 구성되는 지적도면을 이용하였으며, 15개의 시점에 대한 데이터베이스를 생성하였다. 시점간에는 레이어를 구성하는 객체의 약 10%정도에 대하여 공간정보를 변경하였으며, 비공간정보는 고려하지 않았다.

데이터베이스는 다음과 같이 구축하였다. 기법-3에서는 하나의 테이블에 과거와 현재데이터를 모두 저장하였으며, 과거데이터는 기준상태와 비교하여 변화된 공간정보만 저장하였고, 기준상태 정보는 기준상태 메타테이블에 저장하였다. 하나의 테이블에 현재와 과거 데이터를 모두 저장하기 때문에 동일한 객체에 대한 유효시간 범위를 나타내기 위하여 유효시간과 거래시간의 종료시간을 저장하여야 하며, 객체식별자와 유효시작시간으로 구성된 복합키를 주요키로 사용한다. 기준상태 메타테이블은 기준상태의 종료시간을 관리할 필요가 없기 때문에 유효시간과 거래시간의 시작시간만 저장하였다.

객체ID	UNode	DNode	VS	VE	TS	TE
------	-------	-------	----	----	----	----

(a) 공간정보 테이블

객체ID	VS	TS
------	----	----

(b) 기준상태 메타테이블 스키마

(그림 10) 기법-3의 테이블 구성

본 실험에 사용된 하드웨어와 운영체제의 환경은 <표 1>과 같으며, 데스크탑 지리정보시스템인 GEO-

<표 1> H/W와 OS환경

CPU	Pentium-II 333 MHz
Memory Size	128MB
OS	Windows 98

Mania의 질의처리모듈과 자료저장모듈에 3장에서 언급된 함수들을 추가하여 수행하였다.

4.2 데이터 이동 알고리즘을 이용한 과거데이터 저장 시간 분석

기법-3의 과거데이터 저장 방법과 비교하여 (그림 8)에서 설계한 MBNF기법의 데이터 이동 알고리즘에 추가된 부분은 다음과 같다.

첫째, T_n 시점에서 변경이 발생한 객체의 T_{n+1} 시점의 데이터에 유효시간과 거래시간의 종료시간을 추가하는 과정(그림 8의 ①과②)이 추가되었으며, 이 과정의 실행시간은 각각 $O(1)$ 이다.

둘째, 기준상태의 정보와 비교하여 생성된 데이터의 노드 수와 원형상태의 노드 수를 비교하는 과정(그림 8의 ③)이 추가되었으며, 이 과정의 실행시간은 $O(1)$ 이다.

셋째, 새로운 기준상태를 생성하는 과정(그림 8의 ④)이 추가되었으며, 이는 객체의 정보를 표현하는 튜플에서 기준상태를 표현하기 위한 어트리뷰트를 추출하는 과정으로 실행시간은 $O(1)$ 이다.

넷째, 생성된 기준상태를 기준상태 메타테이블에 저장하는 과정(그림 8의 ⑤)이 추가되었으며, 이 과정의 실행시간은 $O(1)$ 이다.

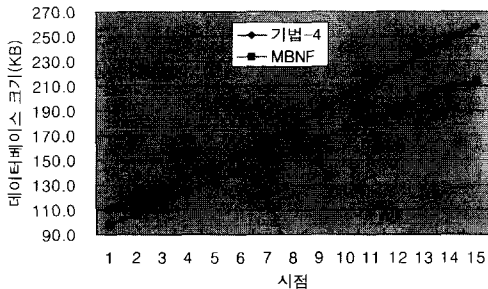
기법-3과 MBNF기법 모두 과거데이터를 저장하기 위하여 가장 많은 시간이 필요한 부분은 기준상태 메타테이블로부터 갱신이 발생한 객체의 기준상태를 검색하는 과정이다. 기준상태 메타테이블에 n 개의 기준상태가 저장되어있고, 트리를 이용한 색인이 구축되어 있다고 가정하였을 경우 기준상태를 검색하는 과정의 실행시간은 $O(\log n)$ 이다. 따라서 기법-3과 MBNF기법의 과거데이터 저장 시간은 각각 $O(\log n)$ 과 $O(\log n) + 5O(1) \approx O(\log n)$ 이다. 따라서 MBNF기법과 기법-3을 이용하여 데이터베이스를 구축하는 시간이 동일한 것으로 분석된다.

4.3 데이터베이스 크기 분석

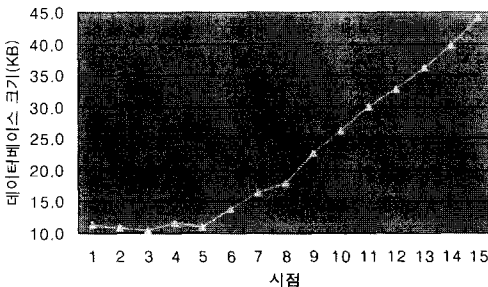
(그림 11)은 각 시점의 데이터베이스 크기를 나타낸 그래프이며, (그림 12)는 두 기법을 이용한 데이터베이스 크기의 차이를 나타낸 그래프이다. MBNF기법을 이용한 데이터베이스의 크기는 현재 데이터베이스의 크기와 과거 데이터베이스의 크기가 포함된 크기이다.

MBNF기법에서는 현재 데이터베이스에 유효시간과

거래시간의 종료시간을 저장하지 않았으며, 기법-3에서는 현재데이터와 과거데이터에 모두 유효시간과 거래시간의 시작 및 종료시간을 저장하기 때문에 시점-1에서의 데이터베이스 크기가 MBNF기법을 이용한 데이터베이스의 크기보다 크게 나타났다. (그림 13)은 동일한 객체에 대하여 시점-15에서 두 기법을 이용하여 저장한 데이터베이스의 예이며, 표현을 단순히 하기 위하여 실제 좌표 대신 숫자를 이용하였다. 그림에서와 같이 기법-3에서는 공간정보의 변경이 누적될수록 변화된 노드들이 중복 저장되지만 MBNF에서는 새로운 기준상태를 설정함으로써 데이터의 중복이 감소되는 것을 볼 수 있다. 따라서 (그림 11)과 (그림 12)와 같이 시점이 진행됨에 따라 기법-3에서는 중복되는 데이터의 양이 증가하기 때문에 MBNF기법보다 증가되는 양이 많은 것으로 나타났다.



(그림 11) 데이터베이스 크기의 변화



(그림 12) (기법-3의 데이터베이스 크기) - (MBNF의 데이터베이스 크기)

4.4 검색 연산시간 분석

본 실험에서는 시간을 만족하는 객체를 우선 검색한 후 이들에 대하여 공간 조건에 대한 비교 연산을 수행

[현재데이터]

객체ID	Node	VS	TS
41	19,20,9,10,17	13	13

[과거데이터]

객체ID	UNode	DNode	VS	VE	TS	TE
41	1,2,3,4,5,6,7,8,9,10,11		1	3	1	3
41	12	2	3	7	3	7
41	13,14	2,3	7	8	7	8
41	1,13,15,16,9,10,17		8	10	8	10
41	18	16	10	13	10	13

(a) MBNF기법

[공간정보데이터]

객체ID	UNode	DNode	VS	VE	TS	TE
41	1,2,3,4,5,6,7,8,9,10,11		1	3	1	3
4133	12	2	3	7	3	7
41	13,14	2, 3	7	8	7	8
41	13,15,16,17	2,3,4,5,6,7,8,11	8	10	8	10
41	13,15,18,17	2,3,4,5,6,7,8,11	10	13	10	13
41	19,20,17	1,2,3,4,5,6,7,8,11	13	now	13	now

(b) 기법-3

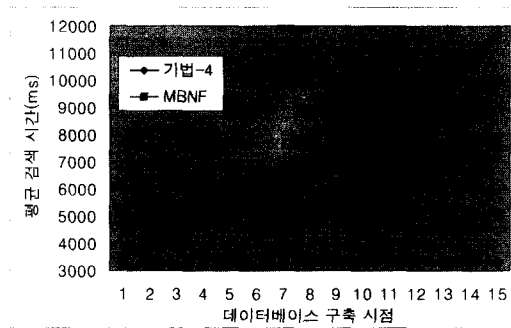
(그림 13) 기법-3과 MBNF기법을 이용하여 저장한 데이터베이스의 예

하였으며, 공간 조건에 대한 비교 연산의 수행 시간은 기법-3과 MBNF기법에서 동일하게 측정되었기 때문에 본 절에서의 분석과정과 (그림 9)의 검색 알고리즘에서 시간만을 고려하였다.

검색 연산시간은 현재시점에 대한 검색과 과거시점에 대한 검색으로 나누어 측정하였으며, 검색하는 시점에 유효한 모든 객체의 정보를 출력하도록 하였다.

현재시점에 대한 검색은 각 시점별로 구축된 데이터베이스를 대상으로 연산을 수행하여 평균 연산 시간을 측정하였다. MBNF기법에서는 현재시점에 유효한 데이터는 모두 현재 데이터베이스에 저장되어 있기 때문에 현재데이터에 저장되어 있는 모든 객체의 정보를 출력한다. 따라서 시점의 변화에 관계없이 상수 시간의 검색 시간이 측정되었다. 기법-3에서는 현재시점에 유효한 객체의 정보를 검색하기 위하여 공간정보 테이블에 저장되어있는 모든 객체에 대하여 유효시간의 범위와 비교 연산을 수행하여야 하며, 검색된 결과를 이용하여 기준상태와의 비교연산을 수행하여야 한다. 따라서 과거데이터의 누적에 의한 데이터베이스 크기의 변화에 많은 영향을 받기 때문에 시점의 변화에 따라

검색 연산의 평균 시간이 선형으로 증가하고 있는 것으로 측정되었다.



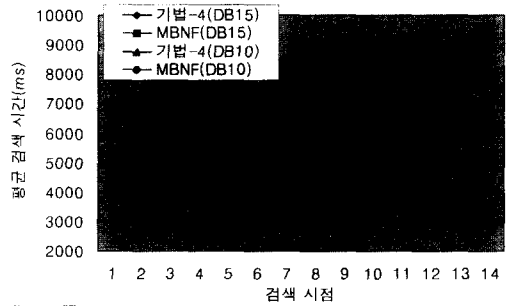
(그림 14) 현재시점에 대한 평균 검색 시간

과거시점에 대한 검색은 시점-10과 시점-15에서 구축된 데이터베이스를 검색 연산의 대상으로 하였다. MBNF기법에서는 두 단계로 수행된다. 첫 번째 단계에서는 현재 데이터베이스에 저장되어있는 튜플들의 유효시간 범위를 비교하여 검색 시점을 포함하는 튜플들을 출력하며, 두 번째 단계에서는 과거 데이터베이스에 저장되어 있는 튜플들의 유효시간 범위를 비교하여 검색 시점에 유효했던 튜플들을 검색한 후 기준상태 메타데이터를 참조하여 각 객체의 기준상태 데이터와 비교 연산을 수행하여 최종 데이터를 생성한다. 첫 번째 단계에서는 기준상태와의 비교 연산을 수행하지 않기 때문에 전체 검색 결과 중에 현재 데이터베이스에 저장되어있는 튜플수가 많을수록 검색 연산은 빠르게 수행된다. 평균 검색 시간은 검색 시점에 유효한 객체의 수와 현재 데이터베이스로부터 검색된 객체의 비율에 영향을 받는다. <표 2>는 시점-15에 구축된 데이터베이스로부터 각 시점에 검색된 튜플의 개수이다.

기법-3에서는 현재시점을 대상으로 수행하는 검색 연산의 방법과 동일한 방법으로 수행된다. 공간정보 테이블에 저장되어있는 모든 객체에 대하여 유효시간의 범위와 비교 연산을 수행하여야 하며, 검색된 결과를 이용하여 기준상태와의 비교연산을 수행한다. (그림 15)에서는 검색 대상 시점이 현재시점에 가까워질수록 검색 평균 시간이 느려지고 있음을 알 수 있다. 이는 검색 시점이 시점-1일 경우 모든 데이터가 기준상태이므로 기준상태의 데이터와 비교하여 검색 시점에 유효한 정보를 생성하는 연산을 수행하지 않기 때문에 비교적

<표 2> 각 시점에 검색된 튜플의 개수

시점	현재 테이블	과거 테이블	현재테이블에서 검색된 튜플수의 비율(%)
1	444	708	38.6
2	454	698	39.4
3	463	680	40.5
4	517	617	45.6
5	535	599	47.2
6	599	526	53.2
7	653	481	57.6
8	6983	454	60.6
9	771	408	65.4
10	816	354	69.8
11	880	299	74.6
12	934	236	79.8
13	1043	136	88.5
14	1134	73	94.0



(그림 15) 과거시점에 대한 평균 검색 시간

MBNF기법으로 검색하는 평균 시간과 근접한 검색 시간을 보여주고 있으며, 검색 시점이 현재시점에 가까워질수록 기준상태의 데이터와 비교하여 검색 시점에 유효한 정보를 생성하는 연산의 수행 빈도가 많아지기 때문에 전체적인 검색 평균시간이 느려지는 것으로 분석된다.

5. 결론 및 향후 연구과제

본 논문에서는 시간 차원을 포함하는 지리정보 데이터를 효과적으로 저장하고 빠른 검색 연산을 수행할 수 있는 분리된 저장구조에 저장하는 MBNF기법을 제안하였다. 제안된 저장구조에서는 객체를 표현하는 데이터의 시간적 특성을 고려하여 현재 데이터베이스와 과거 데이터베이스에 분리하여 저장하도록 하였고, 과거에 유효했던 데이터들은 제안한 MBNF기법을 적용

하여 하나의 공간객체에 대하여 여러 개의 기준상태를 관리하도록 하여 데이터의 중복을 줄일 수 있었다.

본 논문에서 제안한 기법과 기존의 기법에 대한 성능 평가를 요약하면 다음과 같다. MBNF기법을 적용한 데이터베이스에서는 데이터의 유효 시점을 기준으로 분리된 데이터베이스에 저장함으로써 기존의 분리하지 않은 저장구조에서보다 시간을 포함하는 질의 처리가 빠르게 수행됨을 보였다. 또한, 하나의 객체에 대하여 복수의 기준상태를 둬으로써 과거 데이터베이스에 저장되는 데이터의 중복을 최소로 하여 전체 데이터베이스의 크기를 줄일 수 있었다.

본 연구와 관련하여 향후 연구할 과제는 다음과 같다. 본 논문에서는 현재와 과거데이터에 대해서만 고려하였으나 도시계획과 같은 분야에서는 계획도면에 대한 관리를 요구하고 있기 때문에 미래에 유효한 데이터를 다룰 수 있는 저장구조에 대한 연구가 필요하다. 또한 현재 기존의 2차원 지리정보 데이터를 위한 인덱스기법을 확장하여 시간을 포함하는 인덱스에 대한 연구가 진행되어야 한다.

참 고 문 헌

- [1] N. Pissinou, K. Makki, and E. K. Park, "Towards the design and development of a new architecture for Geographic Information Systems," Proc. of the 2nd int. Conf. on Information and knowledge management, pp.565-573, 1993.
- [2] Sudha Ram, J. S. Park, "Modeling Spatial and Temporal Semantics in a Large Heterogeneous GIS Database Environment," Proc. of the 2nd Americas Conf. on Information Systems(AIS '96), Phoenix, Arizona, August 16-18, pp.683-685, 1996.
- [3] Erwig, M., Guting, R.H., Schneider, M., Vazirgiannis, M., "Abstract and Discrete Modeling of Spatio-Temporal Data Types*," 6th ACM Symp. on Geographic Information Systems(ACMGIS '98), pp. 131-136, 1998.
- [4] Erwig, M., Guting, R.H., Schneider, M., Vazirgiannis, M., "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Objects in Databases*," GeoInformatica. 1999.
- [5] C. Claramunt, Theiault, M., "Toward semantics for modeling Spatio-Temporal processes within GIS," SDH, 1996.
- [6] T. Tagashira, T. Amagasa, M. Aritsugi, Y. Kanamori, "Interval-Based Representation of Spatio-Temporal Concepts*," Advanced Information Systems Engineering, 9th Int. Conf., CAISE'97, LNCS 1250, pp. 231-244, 1997
- [7] T. S. Cheng, S. K. Gadia, "A pattern matching language for spatio-temporal databases," Proc. of the 3rd Int. Conf. on Information and Knowledge management, pp.288-295, 1994.
- [8] R. Edsall, D. Peuquet, "Graphical query techniques for temporal GIS," 13th Auto-carto Conf., Seattle, WA, April, 1997.
- [9] P. Dadam, V. Lum, H.-D. Werner, "Integration of Time Versions into a Relational Database System," In Proc. of the Conf. on Very Large DataBases, 1984.
- [10] Langran, G., "Time in Geographic Information Systems," London : Taylor & Francis, 1993.
- [11] May Yuan, "Temporal GIS and Spatio-Temporal Modeling," 3rd Int. Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, New Mexico, January 21-25, 1996.
- [12] A. Tansel, et. al., "Temporal Databases : Theory, Design, and Implementation," The Benjamin/Cummings, 1993.
- [13] Bjørn Skjellaug, "Time and temporal data management -Operationalization in a temporal GIS-," SINTEF Report STF40 A96063, 1996.
- [14] Armstrong, M. P., "Temporality in spatial databases," Proc. GIS/LIS '88, Vol.2, pp.880-889, 1988.
- [15] Langran, G., Chrisman, N. R., "A framework for temporal geographic information," Cartographica, Vol.25, No.3, pp.1-14, 1988.
- [16] Worboys, M., "A Unified Model for Spatial and Temporal Information," The Computer Journal, Vol. 37, No.1. 1994.



박 동 선

e-mail : dseon@netian.com

1992년 인하대학교 전자계산공학과
졸업(공학사)

1994년 인하대학교 대학원 전자계
산공학과 졸업(공학석사)

1996년~현재 인하대학교 대학원
전자계산공학과 박사과정

관심분야 : 데이터베이스, 지리정보시스템, 시간지원 지
리정보시스템



배 해 영

e-mail : hybae@dragon.inha.ac.kr

1974년 인하대학교 응용물리학과
졸업(공학사)

1978년 연세대학교 대학원 전자
계산학과 졸업(공학석사)

1989년 숭실대학교 대학원 전자
계산학과 졸업(공학박사)

1985년 Univ. of Houston 객원교수

1992년~1994년 인하대학교 전자계산소 소장

1982년~현재 인하대학교 전자계산공학과 교수

관심분야 : 데이터베이스, 멀티미디어 데이터베이스, 지
리정보시스템, 실시간 데이터베이스