

□ 기술해설 □

대형 지리 정보 데이터베이스를 위한 객체 지향 GIS 엔진†

한국과학기술원 황규영* · 송주원**

● 목	● 차
1. 서 언 2. 시스템 아키텍처 2.1 GIS 아키텍처의 발전단계 2.2 개발 시스템 아키텍처	3. 공간 액세스 방법 3.1 구석점 변환 기법의 특성 3.2 계층 그리드 화일의 특성 4. 결 언

1. 서 언

지리 공간내에 널리 분포되어 있는 공간 객체들을 관리하는 지리 정보 시스템(Geographical Information System: GIS)은 토지 이용도 평가, 도시 계획, 자원 관리, 시설 관리, 안전 관리, 국방 등의 많은 응용 분야에서 사용되고 있으며, 최근 들어 중요성이 점차 더해가고 있다. 공간 객체(spatial object)는 공간 상의 위치, 모양, 크기, 그리고 다른 공간 객체들과의 상호 관계 등에 대한 속성인 공간 데이터(spatial data)와 공간 객체의 속성중 문자나 숫자로 표현될 수 있는 일반적인 속성인 비공간 데이터(non-spatial data)를 가진다. 따라서, GIS는 공간 객체의 공간 데이터와 비공간 데이터를 유기적으로 관리할 수 있는 구조를 가져야 한다.

데이터베이스 시스템 기술이 세대별로 발전됨에 따라 GIS의 아키텍처도 변화하고 있다[Lar93, Gue94b]. 즉, GIS는 화일 시스템을 기반으로 하는 초기의 형태에서 관계형 데이터베이스 시스템을 기반으로 하는 형태로 발전하였다. GIS에서 관리하는 공간 객체인 지리 객체들은 객체

지향 개념을 이용하여 잘 모델링할 수 있으므로 [Wor94], 최근에는 객체 지향 데이터베이스 시스템을 기반으로 하는 GIS도 활발히 연구되고 있다[Sch91, Dav93, Gue93]. 또한 GIS와 여타 공간 객체를 관리하는 응용들의 요구 사항이 데이터베이스 시스템 기술에 변화를 일으키고 있다. 최근에는 이러한 요구 사항을 고려하여 공간 질의 처리 방법(spatial query processing techniques), 공간 액세스 방법(spatial access methods), 공간 조인 알고리즘(spatial join algorithms) 등에 대한 연구들이 집중적으로 수행되고 있다. 공간 데이터베이스 시스템(spatial database system)은 이러한 연구 내용을 자신의 시스템 아키텍처 내에 반영한다. 공간 데이터베이스 시스템에 대한 현재까지의 연구 내용은 [Gue94b]에 잘 요약되어 있다.

공간 데이터베이스 시스템이란 용어가 정착된 것은 1989년부터 격년제로 개최된 공간 데이터베이스 시스템에 대한 심포지움인 Symposium on Large Spatial Databases(SSD)[Buc89, Gue91, Abe93]에 힘입은 바 크다[Gue94b]. 공간 데이터베이스 시스템은 공간 상에 존재하는 객체들을 관리하는 데이터베이스 시스템이란 의미로 사용되고 있다. 최근 들어 국내외적으로 컴퓨터 연구자들의 공간 데이터베이스 시스템 또는 지리 정보 시스템에 대한 관심은 폭발적으로 늘어나고

†본 문은 한국통신 '95 장기기초 연구과제 "대형 공간 데이터베이스 관리를 위한 GIS엔진의 개발"로부터 부분적으로 지원받았음.

*종신회원

**정회원

있으며 이에 따라 이 분야에 대한 특집들이 여러 학술지들에서 다루어지고 있다[Wha93,한94,Gue 94a].

GIS에서 지원하여야 하는 질의는 크게 공간 질의(spatial query), 비공간 질의(non-spatial query), 복합 질의(mixed-type query)의 세 종류로 나눌 수 있다. 공간 질의는 공간 객체의 공간 데이터에 대한 질의 조건만을 가지는 질의이다. 비공간 질의는 공간 객체의 비공간 데이터에 대한 질의 조건만을 가지는 질의이다. 복합 질의는 공간 객체의 공간 데이터와 비공간 데이터에 대한 질의 조건을 동시에 가지는 질의이다. 이들 질의들을 효율적으로 지원하기 위해서는 액세스 방법들이 필요하다. 비공간 데이터에 대한 액세스 방법을 위해서는 기존의 데이터베이스 시스템 내에서 제공되는 B-트리, B+-트리 등을 사용할 수 있으나, 공간 데이터를 위한 액세스 방법을 위해서는 지리 공간에서 인접한 공간 객체들을 한꺼번에 검색하는 등의 공간 질의 고유의 특성을 고려하는 새로운 방법이 필요하다.

공간 액세스 방법(spatial access methods)은 공간 객체의 위치, 크기, 모양의 관리 방법에 따라 객체 겹침(object overlapping), 객체 분할 또는 중복(object clipping or duplication), 변환(transformation), 공간 순서화(space filling curve) 기법 등으로 분류할 수 있다[Sei87,See88,Ooi91]. 80년대 초반부터 이들 각 부류에 속하는 여러 액세스 방법들이 연구되어 왔으며, 이들은 나름대로의 장단점을 가진다[Lu93]. 따라서, 각 응용에서는 그 응용에 주로 사용되는 공간 객체 및 공간 질의의 특징에 적합한 공간 액세스 방법을 선택하여야 한다.

본 논문에서는 현재 KAIST에서 개발 추진중인 GIS 엔진의 아키텍처에 대하여 소개하고 이를 위하여 연구중인 공간 액세스 방법의 특성에 대하여 설명한다. 이 아키텍처는 공간 데이터베이스 시스템과 객체지향 데이터베이스 시스템 기술을 통합한 시스템을 기반으로 한다. 앞으로 이러한 통합 시스템을 객체지향 GIS 엔진이라 부른다. 이 GIS 엔진 아키텍처에서는 공간 객체 저장 시스템으로 다목적 객체 저장 시스템인 KAOSS(KAIST Object Storage System)[황93]

에 공간 액세스 방법을 추가 확장한 시스템인 GEOSS(Geographical Object Storage System)[송94]를 사용한다. 본 논문의 구성은 다음과 같다. 제 2장에서는 데이터베이스 시스템의 세대 발전으로 변화되어온 GIS 아키텍처 변화 단계에 대하여 설명하고, 객체 지향 GIS 엔진 아키텍처에 대하여 소개한다. 제 3장에서는 GEOSS에서 사용하는 공간 액세스 방법에 대하여 설명한다. 제 4장에서는 연구 결과를 요약하고 앞으로의 연구 방향을 제시한다.

2. 시스템 아키텍처

역사적으로 GIS 아키텍처는 데이터베이스 시스템 기술의 발전에 따라 그 형태를 달리해왔다. 본 장에서는 지금까지의 GIS 아키텍처들을 그 발전 단계에 따라 특징들을 설명하고[Lar93,Gue 94b], 우리가 개발하고자 하는 GIS의 기본 아키텍처를 제시한다.

2.1 GIS 아키텍처의 발전 단계

초기의 GIS는 바로 화일 시스템 위에 만들어 졌다. 이러한 아키텍처를 가지는 시스템의 장단점은 어떤 응용 시스템을 개발할 때 화일 시스템 위에 바로 개발하는 방법과 데이터베이스 시스템을 이용하여 개발하는 두 가지 방법의 장단점과 유사하므로 이에 대한 언급은 여기서는 생략한다. 70년대 초반에 만들어진 GIS들은 이런 아키텍처를 가지는 경우도 있었으나 데이터베이스 기술의 개발과 더불어 최근에는 이런 아키텍처를 가지는 시스템은 거의 없다.

다음 시기에 개발된 대부분의 GIS는 관계형 데이터베이스 시스템을 기반 구조로 하는 아키텍처를 가진다. 이들은 크게 이원형(dual) 아키텍처와 계층형(layered) 아키텍처로 구분할 수 있다.

그림 1(a)의 이원형 아키텍처는 공간 객체의 공간 데이터와 비공간 데이터를 별도의 데이터 관리자를 이용하여 따로 관리하고, 이 두 관리자 위에 통합 관리자를 두어 두 데이터 간의 일관성을 유지할 수 있게 하는 구조를 가진다. 공간



그림 1 관계형 데이터베이스 시스템을 기반으로 하는 GIS 아키텍처: (a) 이원형 아키텍처. (b) 계층형 아키텍처

데이터 관리자로는 관계형 데이터베이스 시스템이 이용되며, 비공간 데이터 관리자는 화일 시스템을 이용하여 독자적인 공간 자료 구조를 유지하는 형태를 가진다. 공간 객체의 두 종류의 데이터 간의 연관성을 유지하게 하기 위해서는 논리적인 포인터(logical pointer) 정보가 유지된다. 현재 사용중인 대부분의 상용 GIS(예, ARC/INFO, SICAD)와 연구중인 일부 시스템(예; GeoQL[Ooi90], SAND[Are91])도 이런 아키텍처를 가진다.

이러한 아키텍처에는 공간 데이터를 표현하는데 독자적인 형태를 사용함으로써 개발되는 GIS 응용에 적합한 고유의 데이터 구조를 나름대로 가질 수 있고 액세스 방법 및 질의 처리에서도 고유의 효율적인 알고리즘을 사용할 수 있다는 장점이 있다. 예를 들어, GeoQL[Ooi90]에서는 skd-tree[Ooi91]를 자신의 액세스 방법으로 사용한다. 이원형 아키텍처에서 하나의 질의는 공간 데이터 부분과 비공간 데이터 부분으로 나누어져 각각의 데이터 관리자에 의하여 처리되어야 한다. 따라서, 질의 처리가 복잡해질 뿐만 아니라 두 부분의 질의 결과를 통합하기 위한 오버헤드도 필요하다. 이외 또 다른 문제점은 두 종류의 데이터를 동시에 다루는 종합적 질의 최적화(global query optimization)가 불가능하다는 점이다.

그림 1(b)의 계층형 아키텍처는 객체의 공간 데이터와 비공간 데이터를 동일 데이터베이스 시스템(예를 들어, 기존의 상용 관계형 데이터베이스 시스템) 내에서 관리한다. 그리고, 공간 데이터를 위한 기능 확장은 이 데이터베이스 시스템 위의 공간 기능 확장기 계층에서 제공한다.

계층형 아키텍처에서 공간 데이터를 표현하는 데는 다음의 두 방법이 있을 수 있다[Gue94b]. 첫째는 객체의 공간 데이터를 고정된 크기의 튜플로 표현할 수 있는 조각들로 나누어 하나의

릴레이션 안에 저장하는 방법이다. 이외의 한 예는 하나의 다각형을 시작점과 끝점의 좌표값들로 표현될 수 있는 선분들로 나누어 하나의 릴레이션 내에 저장하는 것이다. 공간 기능 확장기에서의 공간 데이터 타입을 위한 연산들의 구현을 위해서는 조각들로 표현된 공간 객체의 공간 데이터를 다시 하나의 데이터로 재구성하는 과정이 필요하다. 이 방법의 가장 큰 단점은 재구성 과정에 많은 비용이 필요하다는 것이다. 둘째는 객체의 공간 데이터를 데이터베이스 시스템의 long field를 이용하여 저장하는 방법이다. 이 방법은 첫째 방법보다는 나은 방법이지만, 데이터베이스 시스템은 저장된 공간 데이터를 단지 해석할 수 없는(uninterpretable) 바이트 스트링으로만 간주한다는 문제가 있다. 따라서, 공간 기능 확장기 계층에서 그 내용을 해석하여야 한다. 그러나, 이 아키텍처에서는 z-order[Ore 84] 등의 공간 순서화 곡선(space filling curve)을 이용하여 B-tree 등의 일차원 색인 구조로 공간 객체를 색인하는 제한된 형태의 공간 액세스 방법을 제공하는 장점이 있다.

GIS 아키텍처의 다음 발전 단계는 공간 데이터베이스 시스템을 기반 구조로 사용하는 것이다. 공간 데이터베이스 시스템은 시스템 내에 공간 데이터 타입을 위한 기능들을 제공하는 구조를 가지는 시스템이다. 공간 데이터베이스 시스템의 개발을 위해서는 처음부터 전체를 개발하는 방법과 기존의 데이터베이스 시스템의 소스코드를 이용하여 공간 데이터 타입의 지원을 위한 부분을 확장하는 방법을 고려할 수 있다. 물론 확장형(extensible) 데이터베이스 시스템들을 공간 데이터 타입들을 지원할 수 있도록 확장하는 방법을 사용할 수도 있다.

공간 데이터베이스 시스템에 대한 일반적으로 통용될 수 있는 정의는 아직 내려지지 않은 상태지만, 다음과 같은 기본적 특징은 필수적으로 가져야 한다[Gue94b].

1. 공간 객체의 공간 데이터와 비공간 데이터를 동일 시스템 내에서 관리할 수 있어야 한다.
2. 데이터 모델링과 질의어에서 공간 데이터 타입이 반영되어야 한다.

3. 공간 데이터 타입에 대한 효율적인 액세스 방법과 공간 조인 알고리즘을 제공하여야 한다.

공간 데이터베이스 시스템은 GIS 또는 다른 공간 객체 응용을 위한 기반 구조로서의 데이터베이스 기술을 제공하는 시스템이며 각 응용에 공통적으로 필요한 기본 기능만을 제공한다. 따라서 이 시스템은 어떤 특정한 응용의 GIS를 위한 전체 기능을 모두 제공하는 것을 목적으로 하지는 않는다.

GIS 아키텍처의 또 다른 발전 방향의 하나는 객체지향 데이터베이스 시스템을 기반 구조로 사용하는 것이다. 일반적으로 GIS에서 관리하는 공간 객체인 지리 객체들은 객체지향 개념의 중요한 요소인 상속(inheritance) 개념과 복합 객체(composite object) 개념을 이용하여 상대적으로 쉽게 모델링될 수 있다[Wor94]. 객체 지향 데이터베이스 시스템을 기반 구조로 하는 GIS들에는 O₂를 이용한 시스템[Sch91], GeoO₂[Dav93], GODOT[Gue93] 등이 있다.

객체지향 데이터베이스 시스템을 GIS의 기반 구조로 사용할 경우에는 공간 데이터 타입을 효율적으로 지원하는 데 문제점이 있다. 일반적으로 객체지향 데이터베이스 시스템에는 데이터 타입을 확장하는 기능은 있지만, 공간 데이터베이스 시스템에서 필수적인 요소인 공간 액세스 방법과 질의 처리 방법(예, 공간조인 알고리즘)의 추가 확장을 위한 기능이 없으므로, GIS 응용 개발자에게 이러한 처리 방법을 응용 단계에서 구현하도록 하는 부담을 준다.

2.2 개발 시스템 아키텍처

KAIST에서는 객체 지향 데이터베이스 시스템과 공간 데이터베이스 시스템을 기반으로 하는 GIS들에서 얻을 수 있는 장점들을 동시에 제공하는 객체 지향 GIS 엔진을 개발하고자 한다. 세계적으로도 이러한 특징을 가지는 GIS 엔진의 개발은 태동 단계이다. 상업용 객체 지향 데이터베이스 시스템인 UniSQL에서도 앞으로 공간 액세스 방법을 제공할 예정이다. 본 절에서는 이러한 GIS 엔진 아키텍처를 소개하고 그 특징을

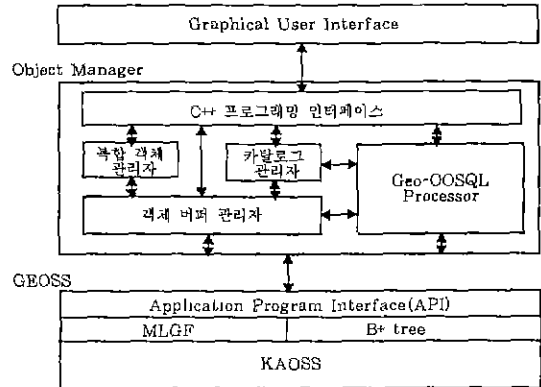


그림 2 객체 지향 GIS 엔진 아키텍처

간략히 설명한다.

그림 2는 우리가 개발하고자 하는 GIS 엔진의 아키텍처이다. 전체 시스템은 크게 공간 객체 저장 시스템, 객체 관리자, 그래픽 사용자 인터페이스 부분으로 구성된다.

공간 객체 저장 시스템인 GEOSS(Geographical Object Storage System)는 다목적 다사용자 객체 저장 시스템인 KAOSS(KAIST Object Storage System)[문94]에 공간 액세스 방법을 추가로 확장한 시스템이다. 현재 KAOSS에서는 액세스 방법으로 B+ 트리를 제공하고 있다. B+ 트리는 비공간 데이터에 대해서는 효과적인 색인 기능을 제공할 수 있으나, 공간 객체간의 근접성(proximity)에 기반을 둔 공간적 선택 질의를 처리하기에는 적합하지 않다. 따라서 이러한 질의를 처리하기 위한 별도의 공간 액세스 방법이 제공되어야 한다. GEOSS에서는 공간 액세스 방법으로는 구석점 변환 기법을 이용하여 객체를 변환하고 이를 다차원 동적 화일 구조의 하나인 계층 그리드 화일(Multilevel Grid File: MLGF)[Wha85,Wha91]을 이용하여 관리하는 방법을 사용한다. GEOSS에서 사용하는 공간 액세스 방법은 제 3장에서 설명한다.

GEOSS는 공간 객체의 액세스를 위한 응용 프로그램 인터페이스(Application Program Interface: API)를 제공한다. 이 API는 B+ tree를 이용하는 비공간 데이터를 위한 부분, MLGF를 이용하는 공간 데이터를 위한 부분으로 구성된

다.

GEOSS는 공간 객체의 모든 데이터를 동일 시스템 내에서 관리하고 공간 액세스 방법을 제공하는 구조를 가지기 때문에 관계형 데이터베이스 시스템을 기반으로 하는 GIS에서 발생하는 문제점들을 해결한다. 이원형 아키텍처에서는 두개의 서로 다른 모델, 서로 다른 인터페이스, 서로 다른 무결성 메카니즘 등에서 기인하는 여러 문제점이 발생한다. 반면에 GEOSS에서는 GIS에서 다루는 모든 데이터들을 함께 관리하므로 단일 모델, 단일 인터페이스, 단일 무결성 메카니즘을 제공할 수 있다. 이원형 및 계층형 아키텍처는 제한된 형태의 공간 액세스 방법만을 제공할 수 있으나, GEOSS는 이러한 제한이 없는 공간 액세스 방법을 제공한다.

본 엔진의 객체 관리자(Object Manager)로는 한국과학기술원 데이터베이스 및 멀티미디어 연구실에 개발중인 객체 지향 데이터베이스 시스템인 ODYSSEUS[박94]의 객체 관리자를 확장하여 사용한다. 이 객체 관리자는 크게 C++ 프로그래밍 인터페이스, 복합 객체 관리자, Geo-OOSQL 처리기, 객체 버퍼 관리자로 구성된다.

C++ 프로그래밍 인터페이스는 응용 프로그램에서 객체 관리자에 접근할 수 있도록 인터페이스를 제공한다. 여기서는 C++ 클래스로써 API를 제공한다.

Geo-OOSQL 처리기(Query Language Processor)는 질의 인터페이스를 통하여 주어진 질의를 처리한다. 이 시스템에서 사용되는 질의어인 Geo-OOSQL은 객체 지향 SQL에 공간 데이터 타입과 공간 연산을 지원할 수 있도록 확장한 형태를 가진다[Ege92,Ege94]. 사용자는 이 질의어를 이용하여 공간 객체에 대한 복합 질의를 구성할 수 있다. 질의어 처리기는 공간 객체 저장 시스템에서 제공하는 액세스 방법을 이용하여 질의어에서 주어진 조건을 만족하는 객체들을 찾는 기능을 한다. 카탈로그 관리자는 질의 수행에 필요한 스키마 정보를 제공해 준다.

복합 객체 관리자는 복합 객체를 이루는 부분 객체들을 하나의 객체처럼 취급할 수 있도록 해준다. 특히 복합 객체가 삭제될 때 부분 객체들도 함께 삭제하는 기능을 수행한다.

객체 버퍼 관리자는 현재 메모리에 존재하는 객체들을 메모리에서 직접 접근할 수 있도록 하여 접근 성능을 높여주는 역할을 한다. 만약 메모리에 객체가 없을 경우 객체를 디스크로부터 읽어서 메모리에 올려 놓는다. 여기서는 응용 프로그램의 가상 메모리 공간을 객체 버퍼로 간주한다.

그래픽 사용자 인터페이스(Graphical User Interface: GUI)는 도시(display) 환경을 정하는 부분, 질의 결과를 화면에 나타내는 부분 등으로 구성된다. 도시 환경은 객체의 데이터를 다른 형태로 화면에 표시하는 데 사용된다. 그리고 이 부분은 추후로 특정 GIS 응용에 필요한 새로운 기능이 추가될 수 있도록 하는 기능을 가진다.

본 절에서는 객체 지향 GIS 엔진의 구성 요소들에 대하여 간략히 설명하였다. 이 GIS 엔진은 한국과학기술원 데이터베이스 및 멀티미디어 연구실에서 앞으로 다년 간에 걸쳐서 개발될 예정이다.

3. 공간 액세스 방법

GEOSS에서는 공간 액세스 방법으로 변환 기법[Hin83,See88]의 하나인 구석점 변환 기법(*corner transformation technique*)을 사용하여 객체를 변환하고, 이들 변환된 점으로 표시되는 객체들을 다차원 동적 화일 구조(*multidimensional dynamic file structure*)의 하나인 계층 그리드 화일[Wha85,Wha91]을 사용하여 저장한다. 제 3.1절에서는 구석점 변환 기법의 특성과 이 특성을 이용한 공간적 선택 질의의 처리 방법에 대하여 설명한다. 제 3.2절에서는 계층 그리드 화일의 특성에 대하여 설명한다.

3.1. 구석점 변환 기법의 특성

구석점 변환 기법은 이차원 원공간(*original space*)에서의 한 객체에 대하여 이 객체를 포함하는 축에 수직인 선들로 이루어지는 최소 포함 사각형(*minimum bounding rectangle*)의 좌하점(*lower-left corner*)과 우상점(*upper-right corner*)의 네 좌표 값을 사차원 변환 공간의 한 점

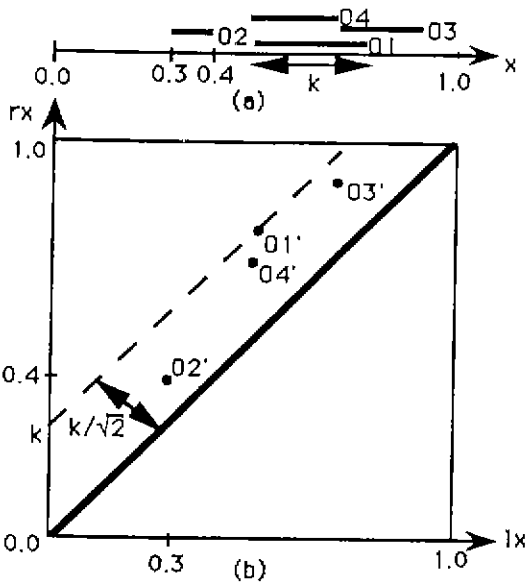


그림 3 구석점 변환 기법의 예: (a) 원공간의 객체들, (b) 변환공간의 객체들.

으로 변환하여 객체를 표현하는 방법이다. 변환된 객체를 색인하기 위해서는 기존에 점 데이터를 색인하기 위하여 제안되어 있는 다차원 색인 구조를 사용한다.

사차원 변환공간을 그림으로 표현하기는 어려우므로, 본 논문에서는 설명의 편의를 위하여 일차원 원공간에서 공간 객체가 길이만을 가지는 경우를 대상으로 논의를 전개한다. 즉, 일차원 원공간 내의 길이 만을 가지는 객체를 구석점 변환 기법을 이용하여 좌측점과 우측점을 변환된 이차원 공간의 가로축인 lx 축, 세로축인 rx 축에 각각 대응시킨다. 원공간의 전체 길이는 1이라 가정한다.

그림 3(a)의 원 공간의 객체들은 그림 3(b)의 윗 부분의 점 객체들로 변환된다. 예를 들어, 객체 O2는 좌측점이 0.3이고 우측점이 0.4이므로 O2'으로 변환된다. 이 방법에서는 모든 일차원 원공간의 객체들은 좌측점의 좌표가 우측점의 좌표보다 작으므로, 변환된 후 이차원 변환 공간에서 대각선의 위 쪽의 점 객체로 존재하게 된다.

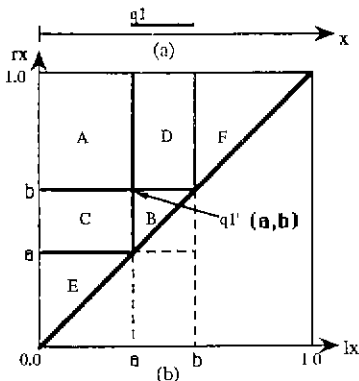
객체의 길이는 변환공간의 점 객체의 위치에서 lx 축으로 수선을 내렸을 때 그 위치에서 대각선과 교차하는 지점까지의 거리와 같다. 따라서, 원공

간에서의 객체의 길이가 길수록 객체는 대각선에서 먼 부분에 위치한다. 원공간의 객체 중에서 길이가 가장 긴 객체(이 그림에서는 O1)의 길이를 k 라고 할 때, 객체가 존재할 수 있는 부분은 대각선을 밑변, 그림의 점선을 윗변, 높이가 $k/\sqrt{2}$ 인 사다리꼴 내부이다. 일반적으로 객체의 길이는 일차원 원공간의 길이와 비교하여 상대적으로 대단히 작으므로, 이 대각선 바로 위쪽의 띠 모양의 부분에만 변환된 점 객체가 분포하게 되어 변환 공간 내에서 객체들은 lx 축과 rx 축의 상관 관계(correlation)가 아주 높은 심각한 불균일(skewed) 분포를 가지게 된다[Hin 83, See88]. 그리드 화일[Nie84], 다차원 신장 해상[Oto84] 등 몇몇 다차원 동적 화일 구조에서는 공간에 객체가 불균일하게 분포하는 경우 디렉토리 크기가 심각하게 커지는 문제가 있으므로[Kim93], 구석점 변환 기법에 적용하기에는 적합하지 않다.

구석점 변환 기법을 사용하기 위해서는 이러한 상관 관계가 높은 불균일 데이터 분포에도 유연하게 대처할 수 있는 다차원 동적 화일 구조가 필요하다[Pag93]. GEOS에서는 이러한 특성을 갖는 다차원 동적 화일 구조로서 계층 그리드 화일[Wha85, Wha91]을 이용한다. 제 3.2절에서는 계층 그리드 화일이 이러한 특성을 가지는 이유를 설명한다.

그림 4(a)의 질의 영역 $q1$ 이 그림 4(b)와 같이 구석점 변환 기법을 사용하여 변환 공간에서 한 점 $q1'$ 으로 변환될 때, 이차원 변환공간에서 객체 존재 가능 영역은 이 객체에 대한 상대적 공간 관계에 따라 $q1'$ 을 기준으로 하여 A에서 F까지의 여섯 개의 영역들로 구분될 수 있다.

각 영역의 특성은 다음과 같다. 영역 A에 존재하는 모든 객체들은 $q1'$ 의 lx 값보다 lx 값(좌측점 좌표값)이 작고 $q1'$ 의 rx 값보다 rx 값(우측점 좌표값)이 크므로, 원공간에서 $q1$ 을 포함한다. 영역 B에 존재하는 모든 객체들은 $q1'$ 의 lx 값보다 lx 값이 크고 $q1'$ 의 rx 값보다 rx 값이 작으므로, 원공간에서 $q1$ 에 포함된다. 영역 C에 존재하는 모든 객체들은 그 lx 값이 $q1'$ 의 lx 값보다 작으며 그 rx 값이 $q1'$ 의 lx 값보다 크고 rx 값보다 작으므로, 원공간에서 $q1$ 의 좌측점과 교차한다. 영역



- A: 원공간에서 q_1 을 포함하는 객체들이 존재하는 영역
- B: 원공간에서 q_1 에 포함되는 객체들이 존재하는 영역
- C: 원공간에서 q_1 의 오른쪽과 교차하는 객체들이 존재하는 영역
- D: 원공간에서 q_1 의 오른쪽과 교차하는 객체들이 존재하는 영역
- E: 원공간에서 q_1 의 왼편에 존재하는 객체들이 존재하는 영역
- F: 원공간에서 q_1 의 오른편에 존재하는 객체들이 존재하는 영역

그림 4 구석점 변환 기법에서의 공간 관계별 영역 구분:
 (a) 원공간 질의 영역 q_1 . (b) 변환 공간에서의 공간 관계별 영역 구분

D에 존재하는 모든 객체는 그 lx 값이 q_1' 의 lx 값보다 크고 rx 값보다 작으며 그 rx 값이 q_1' 의 rx 값보다 크므로, 원공간에서 q_1 의 우측점과 교차한다. 영역 E에 존재하는 모든 객체는 그 rx 값이 q_1' 의 lx 값보다 작으므로, 원공간에서 q_1 의 왼편에 존재한다. 영역 F에 존재하는 모든 객체는 그 lx 값이 q_1' 의 rx 값보다 크므로, 원공간에서 q_1 오른편에 존재한다.

하나의 질의 영역이 결정되면 공간 관계에 따라 변환 공간내에서 처리해야할 영역들이 결정되므로, 이러한 영역 구분은 공간적 선택 질의의 처리에 이용될 수 있다. 예를 들어, 그림 4(a)에서 q_1 과 조금이라도 교차하는 모든 객체를 검색하는 영역 교차 질의(region intersection query)를 처리하기 위해서 검색되어야 하는 영역은 그림 4(b)에서 이 질의 영역을 포함하는 객체들이 존재하는 영역인 영역 A, 이 질의 영역 안에 포함되는 객체들이 존재하는 영역인 영역 B, 이 질의 영역과 교차하는 객체들이 존재하는 영역인 영역 C와 D를 합한 영역이다. q_1' 의 좌표값을 (a,b)라 할 때, q_1 을 질의 영역으로 하는 구간 중첩 질의는 변환 공간에서 ' $lx \leq b$ and $rx \geq a$ '인 조건의 반범위 질의(half range query)로 변환되어 처리된다.

영역 교차 질의를 이차원 원공간으로 확장한 질의인 윈도우 질의(window query) [Kri91]는 이차원 원공간에서 축에 수직인 선들로 이루어지는 질의 영역인 윈도우에 중첩되는 모든 공간 객체들을 검색하는 질의이며, 사차원 변환공간의 반범위 질의로 변환되어 처리될 수 있다. 영역 포함 질의(region enclosure query), 영역 피포함 질의(region containment query)[Kri91]도 유사한 방법으로 처리될 수 있다.

3.2. 계층 그리드 화일의 특성

본 절에서는 구석점 변환 기법을 통하여 변환된 객체를 색인하기 위한 기반 자료 구조로 사용되는 계층 그리드 화일(multilevel grid file) [Wha85,Wha91]의 동적 특성과 구조적 특성에 대하여 설명한다.

계층 그리드 화일은 디렉토리과 데이터 페이지로 구성된다. 디렉토리는 전체 데이터 공간의 분할 상태를 반영하며, 하나의 디렉토리 엔트리는 데이터 공간내의 영역과 일대일 대응 관계를 가진다. 데이터 페이지는 객체들을 저장하는 저장 단위로서 디스크 상에 존재한다. 데이터 페이지는 영역과 일대일 대응 관계를 가지며, 대응되는 영역내에 속하는 객체들만을 저장한다.

계층 그리드 화일은 객체가 데이터 공간에 삽입되고 삭제되는 상황에 따라 분할과 병합을 반복함으로써 동적 변화에 적응한다. 데이터 공간내에 객체가 삽입되는 경우, 객체가 속하는 영역을 찾아 그 영역에 할당된 데이터 페이지에 객체를 삽입하게 된다. 이 결과로 데이터 페이지의 용량이 초과되면(overflow), 해당 영역은 같은 크기를 갖는 새로운 두 영역으로 분할되고 새로운 데이터 페이지가 하나 더 할당된다. 기존의 데이터 페이지에 있던 객체들은 분할된 두 영역의 분할 경계값을 기준으로 각각의 영역에 할당된 두 데이터 페이지에 분산된다.

계층 그리드 화일에서 데이터 공간을 분할할 때 나타나는 가장 큰 특징은 분할이 요구되는 영역만을 분할시키는 부분적 분할 방식(local splitting strategy)[Wha91]을 취한다는 것이다. 이러한 분할 방식은 반드시 필요한 영역만을 생성

시켜 디렉토리 엔트리 수의 증가를 억제하기 위한 전략을 반영한 것이다. 이 결과 계층 그리드 화일의 디렉토리는 저장되는 데이터의 분포나 서로 다른 속성간의 상관 관계 등 데이터 특성에 큰 영향을 받지 않고 삽입되는 객체 수에 선형적으로 비례하여 증가한다[Wha91]. 따라서 계층 그리드 화일은 구석점 변환 기법을 사용하는 경우 발생하는 상관 관계가 높은 불균일 객체 분포에 잘 적응할 수 있으므로 구석점 변환 기법에 적합한 다차원 동적 화일 구조이다.

계층 그리드 화일의 디렉토리 엔트리는 리전 벡터(region vector)와 다음 단계 페이지에 대한 포인터로 구성된다. N개의 속성을 갖는 계층 그리드 화일의 리전 벡터는 N개의 해쉬값으로 구성되며, 해당 디렉토리 엔트리가 나타내는 영역의 위치, 모양, 그리고 크기에 대한 정보를 갖는다. 리전 벡터에서 i번째 해쉬값은 그 디렉토리 엔트리의 영역내에 속하는 모든 객체들의 i번째 속성을 해싱하였을 때 나타나는 해쉬값들의 공통 접두부(prefix)가 된다.

그림 5는 두개의 속성을 가진 계층 그리드 화일에서 분할된 이차원 데이터 공간과 이 데이터 공간의 분할된 상태를 반영하는 디렉토리의 엔트리들을 나타낸 것이다. 디렉토리 엔트리 d1은 첫번째 속성과 두번째 속성을 해싱했을 때 나타나는 해쉬값의 접두부가 각각 '00', '0'인 객체들의 영역을 나타내며, 디렉토리 엔트리 d3은 두번째 속성에는 관계없이 첫번째 속성을 해싱하였을 때 나타나는 해쉬값의 접두부가 '1'인 객체들의 영역을 나타낸다. 즉, 디렉토리 엔트리 d3에서 나타나는 기호 '-'는 두번째 속성 도메인이 아직 분할되지 않은 전체 영역임을 의미한다.

계층 그리드 화일에서는 하나의 영역이 다른

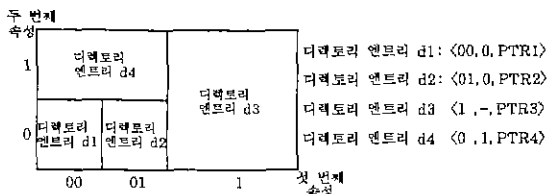


그림 5 디렉토리 엔트리와 영역의 관계

영역에 포함되는지의 여부를 조사하기 위하여 접두부 매칭(prefix-matching)을 사용한다. 접두부 매칭은 한 영역을 나타내는 해쉬값이 다른 영역을 나타내는 해쉬값의 접두부가 되는지 여부를 판단하는 연산이다. 예를 들면, 해쉬값 '101'과 '10100'은 접두부 매칭되고, 해쉬값 '101'과 '10000'은 접두부 매칭되지 않는다. 디렉토리 엔트리 E의 속성 A의 해쉬값이 객체 O의 속성 A의 해쉬값의 접두부가 된다는 것은 디렉토리 엔트리 E가 나타내는 속성 A의 영역에 객체 O가 포함 된다는 것을 의미한다.

다음은 계층 그리드 화일의 디렉토리 구조를 계층 구조로 선택한 이유이다. 전체 디렉토리 엔트리들이 단순히 일차원의 배열 구조로 저장 된다면, 하나의 객체가 어느 영역에 속하는가를 알아보기 위해 최악의 경우 모든 디렉토리 엔트리들을 조사하여야 한다. 계층 구조는 이러한 전체 탐색의 비효율성을 제거하여 준다. 즉, 최하위 단계 디렉토리 D1의 데이터 공간내에서 서로 인접한 영역과 대응되는 디렉토리 엔트리 들을 같은 디렉토리 페이지에 저장하고, 이 디렉토리 엔트리들이 표현하는 영역들을 모두 포함하는 큰 영역과 대응되는 디렉토리 엔트리를 D1의 상위 단계 디렉토리 D2에 유지시킨다. 디렉토리 D2의 디렉토리 엔트리들을 하나의 디렉토리 페이지에 유지할 수 없게 되면, 같은 방식으로 그 상위 단계 디렉토리 D3를 두게되며, 이것은 최상위 단계 디렉토리 엔트리들이 하나의 디렉토리 페이지에 유지될 수 있을 때까지 반복 된다. 따라서 한 객체가 저장된 데이터 페이지를 찾기 위해서는 루트 디렉토리로부터 최하위 디렉토리 D1까지 각 디렉토리 단계에서 하나의 디렉토리 페이지내에 있는 디렉토리 엔트리들을 조사하면서 해당되는 디렉토리 엔트리의 포인터 가 가리키는 페이지들의 체인을 따라 찾아 내려 오면 된다. 이러한 구조적 특성은 계층 그리드 화일을 위한 삽입, 삭제 알고리즘[Wha85,Wha 91]에 의하여 자연스럽게 유지된다.

그림 6(a)는 두개의 속성을 갖는 이단계 계층 그리드 화일의 전체 디렉토리 구조를 표현한 것이다. 그림 6(b)와 그림 6(c)는 각각 그림 6(a)의 디렉토리 D1과 D2가 나타내는 데이터 공간의

4. 결 언

데이터베이스 시스템의 기술이 발전함에 따라 GIS 아키텍처도 변화해왔다. 최근에는 공간 데이터베이스 시스템을 기반으로 하는 GIS와 객체지향 데이터베이스 시스템을 기반으로 하는 GIS가 주로 연구되고 있다. 이들 두 방향은 나름대로의 장점을 가진다.

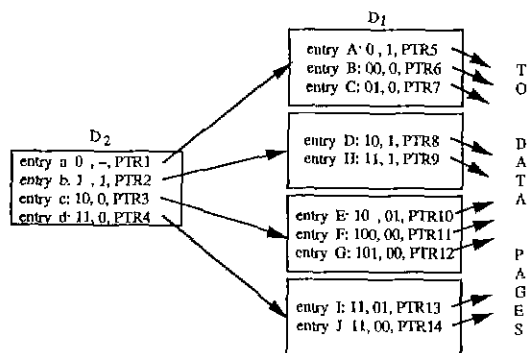
본 논문에서는 KAIST 데이터베이스 및 멀티미디어 연구실에서 앞으로 다년간에 걸쳐서 개발하려고 하는, 객체지향 데이터베이스 시스템과 공간 데이터베이스 시스템을 기반으로 하는 GIS 엔진 아키텍처를 소개하였다. 그리고 이 시스템의 객체 저장 시스템인 GEOSS에서 사용하는 공간 액세스 방법의 특성에 대하여 설명하였다. 본 논문에서 다룬 주요 내용은 다음과 같다.

- 1) 데이터베이스 기술의 발전에 따른 GIS 아키텍처의 발전 단계를 설명하고 각 단계의 특징을 설명하였다.
- 2) KAIST에서 개발중인 GIS 엔진 아키텍처를 소개하고 그 특징을 설명하였다.
- 3) 구석점 변환 기법의 특성을 소개하고, 주어진 질의 영역에 대한 공간 관계별 영역 구분법에 대하여 논의하였다.
- 4) 구석점 변환 기법에 의하여 변환된 객체를 색인하기에 적합한 다차원 동적 화일 구조로서 계층 그리드 화일을 소개하고, 동적 특성과 구조적 특성에 대하여 논의하였다.

향후 연구 방향은 이 아키텍처의 각 부분에 필요한 기술들을 연구하여, 이를 기반으로 완전한 객체 지향 GIS 엔진을 개발하는 것이다.

참고문헌

[Abe93] D. J. Abel and B. C. Ooi(editors), *Proc. 3rd Intl. Symp. on Large Spatial Databases(SSD'93)*, Singapore, 1993.
 [Are91] W. Aref and H. Samet, "Extending a DBMS with Spatial Operations," In *Proc. 2nd Intl. Symp. on Large Spatial Databases(SSD'91)*, pp. 299-318, Aug., 1991.



(a) 이진계 디렉토리 구조.

	000	001	010	011	100	101	110	111
11	A				D		H	
10								
01	B	C			E	I		
00					F	G	J	

(b) D1내의 디렉토리 엔트리들이 나타내는 영역들.

	00	01	10	11
1	a		b	
0			c	d

(c) D2내의 디렉토리 엔트리들이 나타내는 영역들.

그림 6 이차원 계층 그리드 화일의 예

분할 상태를 도면화한 것이다. 그림 6(b)내의 사각형들은 D1의 각 디렉토리 엔트리와 대응되는 영역을 나타낸 것이며, 내부의 문자는 그 영역과 대응되는 디렉토리 엔트리를 의미한다. 따라서 디렉토리 D1에는 열개의 디렉토리 엔트리가 존재한다. D2는 네개의 디렉토리 엔트리를 가지고 있고, D1을 위한 디렉토리인 동시에 루트 디렉토리에 해당한다. D2의 디렉토리 엔트리 c는 첫 번째 속성과 두 번째 속성의 해쉬값의 접두부가 각각 '10', '0'인 객체들이 속한 영역을 나타내며, 이 영역은 D1에서 디렉토리 엔트리 E, F, G가 나타내는 세개의 영역으로 다시 세분된다. 따라서 상위 디렉토리에서 하위 디렉토리로 내려올 수록 보다 구체화된 데이터 공간의 분할 상태가 나타난다.

- [Buc89] A. Buchman, O. Guenther, T. R. Smith, and Y. F. Smith(editors), *Proc. 1st Intl. Symp. on Large Spatial Databases(SSD'89)*, Santa Barbara, 1989.
- [Dav93] B. David, L. Raynal, G. Schoter, and V. Mansart, "GeO2: Why Objects in A Geographical DBMS?," In *Proc. 3rd Intl. Symp. on Large Spatial Databases(SSD'93)*, 1993.
- [Ege92] M. Egenhofer, "Why not SQL!" *Intl. Journal of Geographical Information Systems*, Vol. 6, No. 2, pp. 71-85, Mar.-Apr., 1992.
- [Ege94] M. Egenhofer, "Spatial SQL: A Query and Presentation Language," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 6, No. 1, pp. 86-95, Feb., 1994.
- [Gue91] O. Guenther and H. J. Schek(editors), *Proc. 2nd Intl. Symp. on Large Spatial Databases (SSD'91)*, Zuerich, 1991.
- [Gue93] O. Guenther and W. F. Riekert, "The Design of GODOT: An Object- Oriented Geographic Information System," *IEEE Data Engineering Bulletin*, Vol. 16, No. 3, pp. 4-9, Sept., 1993.
- [Gue94a] R. H. Gueting(editor), Special Issue on Spatial Database Systems, *The VLDB Journal*, Vol. 3, No. 4, Oct., 1994.
- [Gue94b] R. H. Gueting, "An Introduction to Spatial Database Systems," *The VLDB Journal*, Vol. 3, No. 4, pp. 357-399, Oct., 1994.
- [Hin83] K. Hinrichs and J. Nievergelt, "The Grid File: A Data Structure Designed to Support Proximity Queries on Spatial Objects," In *Proc. Intl. Workshop on Graph Theoretic Concepts in Computer Science*, pp. 100-113, 1983.
- [Kim93] S. W. Kim and K. Y. Whang, "Asymptotic Directory Growth of the Multilevel Grid File," In *Proc. Intl. Symp. on Next Generation Database Systems and Their Applications*, pp. 257-264, 1993.
- [Kri91] H. P. Kriegel, "Query Processing in Spatial Database Systems," In book *New Results and New Trends in Computer Science*, Lecture Notes in Computer Science 555, Springer-Verlag, pp. 172-191, June, 1991.
- [Lar93] T. Larue, D. Pastre, and Y. Viemont, "Strong Integration of Spatial Domains and Operators in a Relational Database System," In *Proc. 3rd Intl. Symp. on Large Spatial Databases (SSD'93)*, pp. 53-72, 1993.
- [Lu93] H. Lu and B. C. Ooi, "Spatial Indexing: Past and Future," Special Issue on Geographic Information Systems, *IEEE Data Engineering Bulletin*, Vol. 16, No. 3, pp. 16-21, Sept., 1993.
- [Nie84] J. Nievergelt and H. Hinterberger, "The Grid File: An Adaptable, Symmetric Multikey File Structure," *ACM Trans. on Database Systems*, Vol. 9, No. 1, pp. 38-71, Mar., 1984.
- [Ooi90] B. C. Ooi, *Efficient Query Processing in Geographic Information Systems*, Lecture Notes in Computer Science 471, Springer-Verlag, Aug., 1990.
- [Ooi91] B. C. Ooi, R. Sacks-Davis, and K. J. McDonnell, "Spatial Indexing in Binary Decomposition and Spatial Bounding," *Information Systems*, Vol. 16, No. 2, pp. 211-237, 1991.
- [Ore84] J. A. Orenstein and T. H. Merrett, "A Class of Data Structures for Associative Searching," In *Proc. 3rd Symp. on Principles of Database Systems*, pp. 181-190, 1984.
- [Oto84] E. Otoo, "A Mapping Function for the Directory of a Multidimensional Extendible Hashing," In *Proc. 10th Intl. Conf. on Very Large Data Bases*, pp. 493-506, Singapore, 1984.
- [Pag93] B. U. Pagel, H. W. Six, and H. Toben, "The Transformation Technique for Spatial Objects Revisted," In *Proc. 3rd Intl. Symp. on Large Spatial Databases(SSD'93)*, pp. 73-88, June, 1993.
- [Sch91] M. Scholl and A. Voisard, "Object-Oriented Database Systems for Geographic Applications: An Experiment with O2," In *Proc. Intl. Workshop on Database Management Systems for Geographic Applications*, pp.103-137, Capri, Italy, 1991.
- [Sel87] T. Sellis, N. Roussopolus, and C. Faloutsos, "The R+-tree: A Dynamic Index for Multidimensional Objects," In *Proc. 13th Intl. Conf. on Very Large Data Bases*, pp. 507-518, 1987.
- [See88] B. Seeger and H. P. Kriegel, "Techniques for Design and Implementation of Efficient Spatial Access Methods," In *Proc. 14th Intl. Conf. on Very Large Data Bases*, pp. 10-17, 1988.

[Wha85] K. Y. Whang and R. Krishnamurthy, Multilevel Grid Files, IBM Research Report RC 11516(51719), 1985.

[Wha91] K. Y. Whang and R. Krishnamurthy, "The Multilevel Grid File - A Dynamic Hierarchical Multidimensional File Structure," In *Proc. 2nd Intl. Conf. on Database Systems for Advanced Applications*, pp. 449-459, 1991.

[Wha93] K. Y. Whang (editor), Special Issue on Geographical Information Systems, *IEEE Data Engineering Bulletin*, Vol. 16, No. 3, Sept. 1993.

[Wor94] M. F. Worboys, "Object-Oriented Approaches to Geo-referenced Information." *Intl. Journal of Geographical Information Systems*, Vol. 8, No. 4, pp. 385-399, Jul-Aug., 1994.

[문94] 문양재, 조완섭, 김상욱, 김원영, 이영구, 황규영, "KAIST Object Storage System for Variable-Length Attributes(KAOSS-VLA): 가변길이 속성을 지원하는 관계형 데이터 저장 시스템," 한국정보과학회 '94 가을 학술대회 발표논문집, Vol. 21, No. 2, pp. 7-10, 1994. 10.

[박94] 박종목, 심재균, 이종학, 우준호, 조완섭, 황규영, "ODYSSEUS: UNIX용 다사용자 객체지향 데이터베이스 시스템," 한국정보과학회 '94 가을 학술대회 발표논문집, Vol. 21, No. 2, pp. 31-34, 1994. 10.

[송94] 송주원, 김상욱, 이영구, 황규영, 홍의경, 김장수, "GEOSS: 지리 정보 시스템을 위한 공간 객체 저장 시스템," 한국정보과학회 데이터베이스 연구회지, 제 10권 2호, pp. 24-40, 1994. 8.

[한94] 한기준(editor), 공간 데이터베이스 특집, 한국정보과학회 데이터베이스 연구회지, 제10권 2호, 1994. 8.

황 규 영



1973 서울대학교 전자과 졸업 (B.S.)
 1975 한국과학기술원 전기 및 전자공학과 졸업(M.S.)
 1982 Stanford University(전산학, M.S.)
 1983 Stanford University(전산학, Ph.D.)
 1975 ~1978 국방과학연구소 선임연구원
 1983 ~1990 IBM T.J. Watson Research Center,

Research Staff Member

1991 Visiting Professor, Stanford University
 1992 Visiting Associate Professor, Georgia Institute of Technology
 1993 Hewlett-Packard Laboratories 기술자문(Palo Alto)
 1992 ~1994 한국정보과학회 데이터베이스연구회(SIGDB) 운영위원장
 1993 현재 체신부 통신진흥회의회 DB산업육성분과 위원장
 1995 현재 한국정보과학회 이사 겸 논문지 편집위원장
 Editor: The VLDB Journal
 Editor: Distributed and Parallel Databases: An International Journal
 Associate Editor: The IEEE Data Engineering Bulletin, 1990-1993
 Editor: International Journal of Geographical Information Systems, 1994년-현재
 1990 ~현재 인공지능연구센터 데이터베이스 및 멀티미디어 연구실장
 1990 ~현재 한국과학기술원 전산학과 교수
 관심 분야 : 데이터베이스 시스템, 멀티미디어 데이터베이스, 객체지향 데이터베이스, 하이퍼미디어, GIS, 분산 데이터베이스 및 Client/Server 기술, 연역 데이터베이스, 공학 데이터베이스, 사무자동화, CASE, 전문가 시스템

송 주 원



1981 경북대학교 전자공학과 (전자계산전공) 졸업(B.S.)
 1983 한국과학기술원 전산학과 졸업(M.S.)
 1992 ~현재 한국과학기술원 전산학과 박사과정
 1983 ~현재 한국통신 근무 (선임연구원)
 관심 분야 : GIS, 공간 데이터베이스, 다차원 색인 방법