능동 시스템에서 위치관련 액션 수행을 위한 희소공간 공간객체의 효율적인 영역질의와 최근접질의

김 정 일† · 홍 동 권††

요 약

우리 사회에는 많은 종류의 재해가 발생한다. 일반적으로 재해가 발생하면 귀중한 생명 또는 소중한 재산들을 보호하기 위해 즉각적인 조처가 필요하다. 재해 또는 사고가 발생하면 사고 내용은 즉시 재해 또는 사고 대책 본부에 보고되고 즉각적인 처리를 위한 명령들이 관할 기관에 전달된다. 이 논문은 먼저 능동 데이터베이스의 트리거 기능을 이용한 지능형 무인 비상 대책 시스템을 소개한다. 무인 비상 대책 시스템은 사람의 직접적인 참여 없이 저장하고 있는 지식을 이용하여 다양한 종류의 사고를 처리한다. 이 시스템의 핵심 기능 중 하나는 적절한 처리 명령이 전달될 위치를 신속히 찾아내는 기능이다. 본 논문에서는 무인 대책 시스템과 같은 능동 시스템에서 명령이 전달될 위치를 신속히 찾기 위해서 선형화 방법인 Z-ordering 방법을 사용하여 희소 공간에서 효과적인 영역질의와 최근접 질의를 할 수 있는 새로운 방법을 제시하였다. 뿐만 아니라 희소공간에 존재하는 위치 사이의 거리가 직선거리가 아닌 실제 상황을 고려한 경우에도 처리할 수 있는 방법을 제시하였다.

An Efficient Range Search and Nearest Neighbor Search Algorithm for Action Parts of Active Systems in Sparse Area

Jung-Il Kim† · Dong-Kweon Hong††

ABSTRACT

Various kind of disasters happens in our society. Most of them require immediate treatment to save life or to protect valuable products. When an accident happens in a place, it is reported to the headquarter of emergency measures system. According to the nature of accident several treatments orders are transmitted to the related authorities. In this paper, we introduce an intelligent emergency measures system that uses trigger mechanism of active databases. The system responds to various events spontaneously without intervention of mankind by triggering proper rules. The most important part of an action in the system is the capability of searching places to apply adequate treatments quickly. We have developed a new method for range queries and nearest neighbor queries which utilize the z-ordering technique to get fast responses. Those new methods are further extended to handle more realistic actual distance of road among positions.

키워드 : 능동 데이터베이스(Active database), 공간 데이터베이스(Spatial Database), 영역질의(Range search), 최근접 질의(Nearest neighbor search)

1. 서 론

우리 주변에 널리 사용되고 있는 제어, 감시 시스템들은 부분적으로 자동화가 이루어져 왔지만 여전히 많은 부분이 인간의 직접적인 감시와 참여 속에서 이루어지고 있다. 일반적으로 감시 및 제어 시스템에서 인간의 인위적인 반응은 많은 오류를 발생시키기 쉬우며 기계와는 달리 가용성에 있어서도 많은 한계가 있기 마련이다. 이런 문제를 해결하기 위해 데이터 베이스 내에 지식을 부여하여 사람의 인위적인 감시 및 참여 없이 효율적인 자원의 할당, 정확한 감시 및 처리 절차, 적절한 대처, 최대의 가용성(무인제어의 경우 24시간)등의 기능을 제공하는 무인 제어, 감시 시스템에 대한 연구가 활발히 진행되고 있다[5].

무인 감시, 제어 시스템은 응용에 따라 다양한 형태를 가지는 많은 종류의 신기술을 사용하는 시스템이다. 본 논문은 여러 응용 분야 중에서 사회에서 발생하는 다양한 종류의 사고 및 사건을 적절히 처리하는 기능을 가진 무인 비상 대책 시스템 개발의 일환으로 여러 연구 분야들 중에서 데이터베이스의 능동 기능과 공간 검색 기능을 이용한 실시간 공간 정보 검색 방법을 연구하였다.일반적으로 무인 비상 대책 시스템은 2차원 공간 좌표 X, Y를 가진 임의의 위치에서 발생한 다양한 종류의 사건, 사고에 적절하게 대처하는 능력을 가지면서 효율적인 자원의 할당, 정확한 절

※ 본 연구는 2000년도 계명대학교 비사연구기금으로 이루어졌음.
† 정 회 원 : LG 히다찌
†† 종신회원 : 계명대학교 컴퓨터전자공학부 교수
논문접수 : 2000년 6월 19일, 심사완료 : 2001년 3월 30일

차의 수행, 최대의 가용성(24시간 가용)을 보장하는 능동 실시간 시스템이다. 어떤 임의의 장소, 임의의 시간에 사건이 발생할 경우, 예를 들어 화재가 발생한 경우 감지된 사건이 목격자의 신고 또는 화재 경보기와 같은 다양한 방법으로 무인 비상 대책 시스템으로 전달되고, 전달된 사건은 무인 비상대책 시스템의 데이터 베이스에 기록된다. 발생한 사건이 기록되는 순간 사건의 종류에 따른 적절하고 신속한 조치를(예를 들면 화재 발생 지점에서 제일 가까운 소방서에 고층 사다리가 있는 소방차를 출동시키라고 명령) 취하기 위해 지식베이스에서 적절할 지식을 찾아 수행할 액션을 파악한다. 그리고, 해당 액션이 가해질 위치를 공간 검색 방법을 이용하여 찾아내어 적절한 위치에 액션을 취하게 되며, 취해진 액션과 사건은 앞으로의 사건 및 사고 발생을 줄이기 위한 예방책으로 사건 및 사고의 유형을 예측하기 위한 자료로 사용하기 위해 데이터베이스에 기록된다.

본 논문에서는 데이터베이스에서 액션이 가해질 위치를 (예, 화재 지점에서 고가 사다리 소방차를 보유하고 있는 가장 가까운 소방서) 신속히 찾기 위한 방법을 연구한다. 특히 가상적인 무인 비상 대책 시스템의 구현[7-8]에서 사용될 여러 가지 건물들의 위치 정보를 표현하는 방법과 비상 대책 시스템에서 가장 빈번히 사용되는 연산인 영역질의(Range Search)와 최근접 질의(Nearest neighbor search)를[5] 효과적으로 수행하기 위한 방법을 제안한다.

2. 관련 분야 연구

2.1 무인 비상 대책 시스템

무인 감시 및 제어 시스템은 다음과 같은 분야에 적절하게 사용될 수 있다.

- 사람이 직접 참여하기 힘든 우주, 해저, 위험한 장소의 감시 및 제어
- 병원의 환자 모니터링과 같이 제어할 대상이 너무 많아서 직접 사람이 일일이 참여하기 힘든 환경.
- 항공 제어, 통신 및 전력의 감시 및 제어 등과 같이 제어 절차가 복잡하며 신속히 이루어져야 하는 환경.
- 미국의 911 dispatcher 시스템과 같이 무인 비상대책 시스템으로서 많은 양의 요구에 대해서 24시간 내내 신속한 처리를 계속 지원해야 하는 환경

위와 같은 종류의 무인 감시 및 제어 시스템 중에서 많은 양의 요구를 24시간 내내 신속하게 처리 해줄 수 있는 무인 비상대책 시스템은 일반적으로 다음과 같은 시나리오에 따라 동작한다.

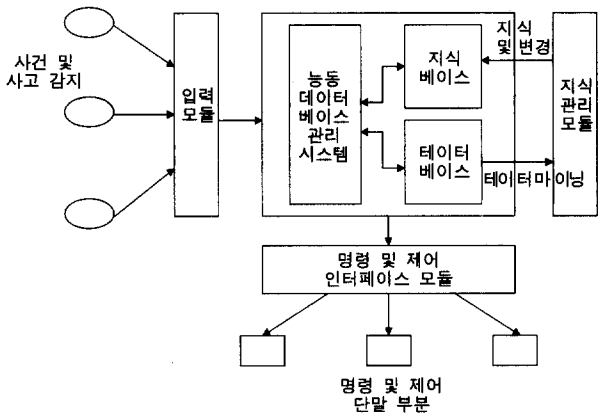
(1) 임의의 장소, 임의의 시간에 사건이 발생, 예를 들어 화재가 발생했다.

(2) 신고 또는 화재 경보와 같은 다양한 방법으로 화재는 무인 비상대책 시스템으로 전달되고, 전달된 사건은 무인 비상대책 시스템의 데이터 베이스에 기록된다.

(3) 기록된 사건의 종류에 따른 적절하고 신속한 조치를 취하기 위해 지식베이스에서 적절할 지식을 찾아 수행할 액션을 파악한다(지식의 예, 사고 지점에서 제일 가까운 소방서와 응급실이 있는 병원에 고층 사다리가 있는 소방차와 환자 응급차를 출동시키라고 명령).

(4) 수행될 액션에서 액션이 적용될 위치(예를 들면 사고 지점과 제일 가까운 소방서와 병원)를 공간 검색 방법을 이용하여 찾아내어 액션을 수행하게 된다.

(5) 수행된 액션과 사건은 앞으로의 사건 및 사고 발생을 줄이기 위한 예방책 및 지식의 증가 및 변경 시 자료로 사용하기 위해 데이터베이스에 기록된다.



(그림 1) 무인 비상대책 시스템의 전체 구조도

(그림 1)은 전형적인 무인 비상 대책 시스템의 개략적인 모습을 나타낸다. 사건 및 사고 감지가 다양한 종류의 센서 또는 여러 가지 방법으로 이루어지면 능동 데이터베이스 관리 시스템은 지식 베이스의 내용을 이용하여 적절한 반응을 명령 및 제어 인터페이스 모듈로 전달한다. 이때 명령 및 제어가 전달되어야 할 곳을 신속히 파악하기 위하여 위치 정보의 적절한 관리 및 검색 기능이 필요하다.

2.2 2차원 공간에서 위치 표시 방법

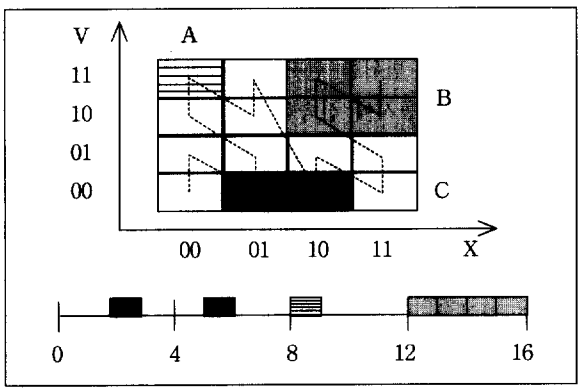
일반적으로 공간 객체 정보는 위치, 모양, 넓이 등의 다양한 속성을 가지고 있으며 응용 분야에 따라 속성의 중요도는 서로 달라진다. 무인 비상 대책 시스템과 같은 형태의 응용에서는 모양 및 넓이 등의 다른 속성은 중요한 역할을 하지 못하며 위치 속성이 가장 중요한 의미를 가진다. 2차원 공간의 위치를 데이터베이스에 표시하는 방법을 살펴보면 다음과 같다.

먼저 공간 좌표 X, Y를 테이블의 필드로 표시하는 방법이다. 우리가 사용하는 상용 데이터 베이스에서는 좌표로 구성된 공간 객체에 영역 질의 연산을 수행할 때 빠른 검색을 위해서 인덱스를 사용하며 이때 저장된 데이터가 연

속적일 때 데이터 검색이 효율적으로 이루어진다[6]. 하지만 공간 좌표 X와 Y를 테이블의 필드로 표시하고 B-tree를 이용한 결합 인덱스를 생성할 경우 공간적으로는 가깝지만 인덱스 자료구조에서는 떨어져 있으므로 검색의 효율성이 떨어진다. 예를 들어, 좌표(2, 3)과 좌표(3, 3)을 인덱스를 사용하여 연속적으로 검색한다고 가정할 경우 많은 좌표(2, Y)의 형태 때문에 좌표(3, Y) 형태의 데이터를 위한 인덱스는 같은 블록 내에 존재하지 않게 된다. 따라서 공간적으로 이웃한 좌표들을 검색할 때 디스크 IO가 더 일어나야 하므로 검색시간이 느려지고, 따라서 반환시간이 느려진다.

다른 방법은 2차원의 위치정보를 1차원의 선형으로 변환하는 기법을 사용하여 저장하는 것이다. 이 경우 공간적으로 연속된 많은 것들이 디스크에 연속적으로 할당될 확률이 상당히 높다. 선형 변환기법에는 Z-Ordering이나 Hilbert Curve, Gray Code[5] 같은 방법들이 있다. 이 가운데, Z-Ordering 기법은 현재 소개된 변환 기법 중 가장 간단하며 본 논문의 검색 방법에 채택된 것이므로 소개한다[5, 6, 13, 15].

(그림 2)는 Z-Ordering에 대한 개념도 이다.



(그림 2) Z-Ordering 개념도

Z-ordering에서 2차원 정보의 1차원 mapping 방법은 bit interleaving을 사용하여각 축의 값을 2진수로 변환하고 이를 결합한 후 다시 10진수로 변환하면 Z값을 얻을 수가 있으며, 좌표 A의 Z값을 구하기 위한 설명이 (그림 3)에 나타나 있다.

$$Z_A = \text{Shuffle}('1,2,1,2', x_A, y_A)$$
$$= \text{Shuffle}('1,2,1,2', 00, 11) = 0101 = (5)_{10}$$

(그림 3) Shuffling을 이용한 Z 값 계산 방법

(그림 3)에서 Bit interleaving 방법을 나타내는 Shuffle('1, 2, 1, 2)의 의미는 X좌표의 값과 Y좌표를 나타내는 2진수를 서로 섞을 때 1은 X좌표를 나타내며 2는 Y좌표를 나타낸다. 이렇게 생성된 Z값은 (그림 2)의 진행 방향과 같이 Z값이 비슷할 경우 실제 좌표가 인접해 있을 가능성이 높다.즉, 공간 객체 질의에서 Z값을 이용하면 변환 과정 없이 저장한 것보다 공간 객체 검색 수행 시간이 빨라 질 수 있는 이점이 있다. 이러한 Z값을 이용하여 영역 질의(Range Query)와 최소 근접 질의(Nearest Neighbor Query)를 하게된다.

- 영역 질의(Range Query) : 임의의 지점에서 일정한 영역 내 존재하는 모든 데이터를 반환해 주는 질의.
- 최소 근접 질의(NearestNeighbor Query) : 임의의 지점에서 제일 가까운 곳에 위치하는 데이터를 반환하는 질의.

또 다른 방법은 공간 객체 검색을 위해 R-Tree나 Quad Tree 등의 다차원 검색 인덱스[3, 7, 10, 11]를 사용하는 방법이다. 이러한 다차원 검색 인덱스들은 길이, 다각형, 원, 사각형 등의 다양한 형태의 공간 객체에 관계된 많은 종류의 특별한 질의들을 효과적으로 처리하는 기능을 제공한다. 하지만 이 분야에 대한 많은 연구들은 빠른 응답보다는 다양한 종류의 공간 연산들에 대한 정의와 효과적인 공간 연산 구현 방법에 대해서 연구하고 있다[17]. 게다가 R-tree와 같은 다차원검색 인덱스는 비 공간 자료인 일반자료와 결합한 인덱스를 효과적으로 구현하기가 어렵다. 2차원 좌표(X, Y)와 좌표에 위치하고 있는 객체의 분류(소방서인지 또는 경찰서인지) 즉 3개의 속성을 지닌 것을 3차원 정보로 표시하고 다차원 검색 인덱스로 검색하는 것이 상당히 효과적이지 못하다. 즉 객체의 분류와 단순한 위치에 의한 영역질의와 최근접 질의를 효과적으로 지원하기에는 R-tree와 같은 자료구조는 너무 부담이 된다.

본 논문에서는 공간 객체의 위치 정보와 공간 객체의 성격(예를 들면 소방서, 경찰서, 병원 등을 나타냄)을 결합 인덱스로 표현하기 적합하면서도 인덱스 구조의 검색 방법이 다차원 인덱스보다 간단한 B-Tree 구조[16]를 사용하여 무인 비상 대책 시스템을 구현하는 과정에서 자주 발생하는 공간 연산의 검색 알고리즘을 제안한다.

3. 공간 정보 검색 알고리즘

3.1 영역질의

영역 질의[5]는 임의의 지점에서 반경 R로 만들어지는 일정한 영역 내 존재하는 공간 객체를 반환해 준다. 예를 들면 "X, Y에서 반경 500 미터 이내의 병원들을 찾아라"하는 명령이 있을 때 각 좌표간의 거리는 유클리디언 거리를 사용한 기존의 방법들을 보면 다음과 같다.

3.1.1 방법 1 - 완전 서치(Exhaustive Search)

```
Exhaustive_search1(X, Y, Rad)
{
   posx = X, posy = Y ;
   while (scale * √((X − posx)² + (Y − posy)²) <= 500)
   {
         posx, posy에 병원이 있는지 인덱스를 이용하여 검사 ;
         posx 또는 posy를 원 중심에서 바깥 방향으로 증가 ;
   }
   return 결과 ;
}
```

3.1.2 방법 2 - 완전서치(2)

```
Exhaustive_search2(X, Y, Rad)
{
    // 종류, 위치 Z값으로 결합 인덱스가 생성되어 있다.
    인덱스를 이용하여 찾으려고 하는 것의 종류에 해당하는 것의
    첫 번째 인덱스 위치를 찾는다.
    for(현재 인덱스에서부터 종류의 마지막까지 순차적으로 인덱
      스를 읽으면서)
    {
        각 인덱스 아이템의 위치 Z 값을 좌표 값으로 다시 변환
        (posx, posy) ;
        dist = scale * √ (X−posx)² + (Y−posy)² ;
        if (dist <= Rad)
              결과를 저장 ;
    }
    return 결과 ;
}
```

기존의 검색 방법인 방법 1과 방법 2는 완전서치(Exhaustive search) 방식으로 필요한 답을 찾기 위해서 검색 해야 할 공간이 너무 커서 무인 비상 대책 시스템의 개발 환경에서 적합하지 않음을 발견할 수 있다. 무인 비상대책 시스템의 환경을 살펴보면 검색 영역이 결과에 비해서 상대적으로 매우 넓다. 즉 반경 R로 만들어지는 검색 영역 내에 검색 목표의 분포도가 굉장히 희박하다. 예를 들어서 무인 비상대책 시스템에서 주된 역할을 할 소방서, 경찰서, 종합 병원 등을 살펴보면 일반적으로 상당히 넓은 지역에 드물게 분포되어 있다. 본 논문에서는 이런 공간을 ***희소 공간***이라 한다.

● 희소 공간에서 방법1의 평가

먼저 반경 R로 만들어지는 영역 안의 좌표 개수가 N이라 하자. N의 값은 응용에서 사용하는 좌표의 촘촘함 정도에 따라 달라지지만 영역질의에서 상당히 큰 값을 갖는다. 예를 들어 각 좌표의 기본 스케일이 1 미터라 할 경우 반경 500 미터 안에는 수십 만개 이상의 좌표가 생긴다. 이 경우 검색 방법은 위치, 종류의 결합 인덱스에 대해서 위치 값을 Z 값으로 변환한 후 Z 값에 있는 것이 원하는 종류인지 검사한다. N 번의 인덱스 검색이 요청될 경우 각각의 인덱스들은 순차적으로 검색이 불가능하므로 독립적인 N 번의 B-tree 검색 요청이 이루어지며 B-tree의 깊이가 3이라 할 때 최고 3N의 디스크 IO가 발생한다.

● 희소 공간에서 방법2의 평가

방법 2의 경우 종류, 위치 Z 값으로 결합 인덱스가 이루어져 있다. 데이터베이스에 저장되어 있는 검색 대상의 종류에 해당하는 것의 개수가 M이라 하자. M의 값은 검색 대상의 종류와 무인 비상대책 시스템이 관할하는 영역의 넓이에 따라 편차가 심하다. 만약 검색 대상이 소방서 또는 종합 병원이며 관할 영역이 광역시(예, 대구시의 경우) 정도가 된다면 M의 값은 10에서 50 정도의 값을 가질 것이며 검색 대상이 파출소와 같은 경우 수백 또는 수천이 될 수도 있다. 이 경우 검색 방법은 종류의 값을 키로 하여 인덱스에 있는 첫 번째 것을 찾은 후 순차적으로 M개를 모두 검사한다. 검사하는 방법은 각각의 Z 값을 읽은 후 Z 값을 X, Y 좌표로 바꾼 후 중심 지점으로부터 거리를 계산한 후 그 값이 R 이하인지를 검사한다.

앞의 2가지 방법을 비교해 볼 때 무인 비상 대책 시스템과 같은 희소 공간에서는 방법 2가 훨씬 더 효과적임을 알 수 있다. 하지만 방법 2의 경우도 여전히 검색 해야할 대상이 M으로 너무 많아 무인 비상 대책 시스템에 사용하기에는 적합하지 않다.

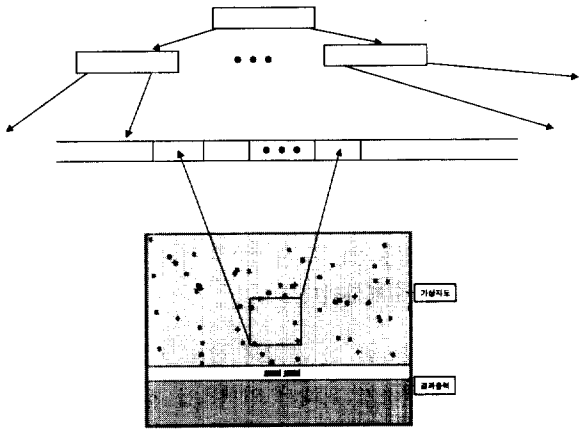
3.2 희소 공간을 위한 검색 방법

무인 비상 대책 시스템은 주어진 사건에 대해서 저장된 지식을 이용하여 정확한 절차를 수행해야 한다. 하지만 절차에 따라 명령을 수행하는 장소를 결정할 때 지나치게 많은 시간이 소요된다면 비상 대책 시스템으로서의 기능을 상실하게 된다. 본 논문에서는 희소 공간에서 Z-ordering의 특성과 원 영역을 포함하는 최소 크기의 정사각형을 이용하여 검색 대상의 범위를 현저하게 줄여서 정확한 위치 연산을 수행하면서 동시에 신속히 수행하는 방법을 제안한다. 먼저 여러 가지 선형 변환 기법 중에서 Z-ordering의 특성을 살펴본다. 선형 변환 기법에는 Z-ordering외에 디스크 IO 관점에서 더 좋은 성능을 보이는 Hilbert Curve 등의 방법이 있다[5]. 선형 변환 기법의 성능은 공간적으로 인접해 있는 다차원 공간의 좌표들이 선형화 되었을 때 어느 정도 인접해 있는가를 측정하여 결정한다. Hilbert Curve가 선형화 하는 모습에 급격한 움직임이 없어 더 좋은 선형화 방법으로 평가되고 있다. 하지만 선형 변환 기법 중에서 Z-ordering은 Hilbert Curve보다 선형화 모양에 급격한 변화가 많아 공간적으로 인접한 좌표들이 선형화 되었을 때 서로 떨어지는 지는 것들이 더 많지만 다음과 같은 우리가 원하는 특성이 존재한다.

Z-ordering 값을 이용한 2차원 공간에서 사각형의 검색 영역을 설정했을 때

(1) 사각형의 왼쪽 아래 모서리 좌표의 Z 값 Z_1은 사각형 내에 존재하는 모든 좌표의 Z 값보다 작다.
(2) 사각형의 오른쪽 위 모서리 좌표의 Z 값 Z_2는 사각형 내에 존재하는 모든 좌표의 Z 값보다 크다.

위의 2가지 특성으로 영역 내에 존재하는 개체의 좌표 Z 값 Z_n은 항상 Z_1 보다 크고 Z_2 보다 작다는 사실을 발견할 수 있다. 이 사실을 바탕으로 다음과 같은 방법을 제안한다.



(그림 4) 사각형과 B-tree를 이용한 표현

제안하는 방법(range_search)
1. 위치 좌표를 여러 가지 선형 알고리즘 중에서 z-ordering을 사용하여 선형화 한다.
2. 목표의 종류와 선형화된 위치 값으로 이루어지는 결합 인덱스를 구성
3. 반경 R로 구성된 원 모양의 영역을 포함하는 최소의 정사각형 영역을 구성한다.
4. 사각형 왼쪽 하단 모서리 좌표와 오른쪽 상단 모서리의 Z 값을 Z1, Z2로 변환한다.
5. 종류와 Z1값을 이용한 결합인덱스에서 Z1보다 크거나 같은 것을 찾는다.
6. Z2값 보다 큰 값을 만날 때까지 순차적으로 검사한다.
7. 검사하는 방법은 위치 Z 값을 읽은 후 Z 값을 X, Y 좌표로 바꾼 후 중심 지점으로 부터 거리를 계산한 후 그 값이 R 이하인지를 검사한다.

```
range_search(x, y, r, category) //좌표 x, y 반경 r, 종류 category
{
        z1 = z_linear(x, y) ;
        left_low_x = x - r ;
        left_low_y = y - r ;
        right_upper_x = x + r ;
        right_upper_y = y + r ;
        z2 = z_linear(left_low_x, left_low_y) ;
        z3 = z_linear(right_upper_x, right_upper_y) ;
        search_item = category, z2 ;

        DECLARE LOC_CUR CURSOR FOR
        SELECT location
        FROM table
        WHERE category = category AND
                (location >= Z1 AND location <= Z2) ;

        OPEN LOC_CUR ;
        while(1)
        {
                FETCH LOC_CUR INTO : z_result ;
                if (sqlca.sqlcode == NOMORE)
                        break ;
                else
                {
                        reverse_linear(z1, x1, y1) ; // Z1값을 이용
                                                     하여 x1과 y1 계산
                        if (r <= u_distance(x, y, x1, y1))
                                result[i++] = z_result ;
                }
        }
        CLOSE LOC_CUR ;
        return(result) ;
}
```

3.2.1 희소 공간에서의 영역질의 평가

(1) 제안한 방법의 정확성

2차원 공간 좌표를 Z-ordering 방법을 사용하여 Z 값으로 변화한다. 이때 사용하는 Bit Interleaving 방법에서 X, Y 좌표의 이진 표현을 서로 섞을 때 앞에서부터 1비트씩 교대로 섞으면 특성상 X 좌표의 값이 같으면 Y 좌표 값이 큰 것이 Z 값이 크게 된다. 같은 방식으로 Y 좌표 값이 같으면 X 좌표 값이 큰 것이 Z 값이 크게 된다. 따라서 사각형 내부에 존재하는 모든 좌표의 Z 값은 항상 왼쪽 하단 모서리의 Z 값 보다 크고 오른쪽 상단 모서리의 Z 값보다는 작다.

(2) 효율성

영역질의를 수행하기 위해서 먼저 원 영역을 포함하는 최소의 정사각형 영역을 구성하고 정사각형의 왼쪽 하단 모서리와 오른쪽 상단 모서리 Z 값을 구한다. 각각의 값을 Z1과 Z2라 할 때 B-tree의 검색에서 (그림 4)와 같이 Z1과 Z2 사이의 값으로 검색 범위를 제한하여 원 영역에 포함되는 개체를 신속히 찾을 수 있다. 이때 원 영역 외의 개체 중에서 Z1과 Z2 값 사이에 존재하는 것이 있을 수 있다. 하지만 Z1과 Z2 사이에 존재하는 값은 많지만 희소 공간에서는 B-tree가 개체의 종류와 위치 Z 값의 결합 인덱스이므로 종류가 같으면서 Z1과 Z2값 사이에 존재하는 인덱스 값을 가진 것은 거의 존재하지 않거나 아주 작은 개수가 존재한다. 검색 방법의 효율성은 영역질의를 수행하기 위해서 걸리는 시간으로 측정할 수도 있지만 본 논문에서는 검색에서 실제로 읽어온 디스크 IO의 횟수로 평가한다. 따라서 본 논문에서 제안한 방법은 희소 공간에서 효율적인 질의가 가능하다.

3.3 희소 공간에서의 최근접 질의

최근접 질의는 "사건 및 사고 발생지점 X, Y에서 가장 가까운 종합병원을 찾아라."와 같은 형태로 발생된다. 제안하는 최근접 질의 함수 nearest_neighbor를 수행하는 방법은 다음과 같다.

제안하는 방법(nearest_neighbor)
- 먼저 사건지점 X, Y에서 제일 가까운 병원을 찾기 위해서는 병원이라는 종류와 X,Y로 만들어지는 Z1 값을 가지고 B-tree를 검색하여 Z값으로 제일 가까운 Z2값을 찾는다.
- 그 다음 X, Y에서 Z2까지의 거리를 반경으로 하는 영역 질의를 앞의 방법을 수행하여 병원을 찾아낸다.
- 찾아낸 병원 중에서 제일 가까운 병원을 찾는다.

```
nearest_neighbor(x, y, category)
{
        z1 = z_linear(x, y) ;
        search_item = category, z1 ;

        SELECT MIN(loc) into : z2
        FROM table
        WHERE category=category AND loc >= z1 ;
```

```
        reverse_linear(z1, x1, y1) ; // Z1값을 이용하여 x1과 y1을
                                        구해낸다.
        r = u_distance(x, y, x1, y1) ;
        result = range_search(x, y, r, category) ;
        return(min(result)) ;
}
```

3.4 희소 공간 영역질의 알고리즘의 확장

우리사회의 일반적인 운반 및 이동 매체는 비행기나 헬리콥터가 아니라 자동차이다. 따라서 모든 이동은 도로를 따라 이루어지며 영역 질의 및 최근접 질의에서도 도로의 상태를 고려한 계산이 이루어져야 한다. 도로의 상태는 실제 도로를 따라 이동하는데 소요되는 시간에 비례하는 값으로 표시되어야 하는데 시시각각으로 변화하는 교통혼잡도, 교차로 수, 차선 수 등과 같은 많은 변수들 때문에 적시에 정확한 값을 산출하기가 쉽지 않다[1, 2]. 그러므로 본 논문에서는 2 지점을 이동하는데 소요될 예상 시간에 비례하는 값을 도로가 연결된 모든 2 지점 사이에 표현하고 앞에서 제안한 영역질의와 최근접 질의를 확장한다. 이때 2 지점간의 직선거리는 2 지점 사이에 도로가 존재하며 전혀 막힘이 없는 가장 좋은 상태에서 이동에 소요되는 시간에 비례하는 값이 된다. 임의의 지점 X, Y에서 가장 가까운 곳을 찾는 방법은 다음과 같다.

```
실제거리 최근접 질의 방법(p_range_serach)
1. 먼저 X,Y를 Z1으로 변환한다,
2. B-tree상에서 Z1값보다 큰 값 Z2와 작은 값 Z3을 구한다.
3. Z2와 Z3 중에서 Z1의 위치와 직선거리가 가까운 것을 선택한
   다. (Z2라 하자)//Z2가 실제 거리상 제일 가까운 것이기를 희망
   하면서
4. Z1과 Z2 사이의 실제 거리를 계산하여 그 값을 R이라 한다.
5. Z1에서 직선거리 반경 R내의 검색 대상 종류에 속하는 위치들
   을 찾는다.
6. 반경 R내에 해당 종류가 아무 것도 없으면 Z2가 실제 거리상
   가장 가까운 것이다. 하지만 반경 R내에 존재하는 것이 있으면
   실제거리를 계산하여 비교한다.
7. 최소 값을 찾아낸다.
```

```
p_range_search(x, y, r, category)
{
        result = range_search(x, y, r, category) ;
        // 직선거리 500 이내의 병원 위치가 result에 저장
        result에 있는 각각의 병원 위치에 대해서
        shortest path algorithm을 적용하여 실제 500 이내에 있는
        것을 선택
}
```

4. 결 론

우리 사회에 존재하는 많은 종류의 데이터를 보면 위치 정보를 가진 것들이 상당히 존재한다. 예를 들어서 이동 통신용단말기들은 계속적으로 위치가 변화하기 때문에 효과적인 통신을 위해서는 위치 정보의 사용이 필수적이다. 본 논문에서 사용하는 무인 비상 대책 시스템의 경우도 다양한 종류의 사건 및 사고에 효과적인 대응 방안을 제시하기 위해서 위치 정보를 사용한 최근접 질의와 영역질의를 수행한다. 본 논문에서는 무인 비상 대책 시스템과 같이 위치 정보를 많이 사용하는 시스템에서 가장 빈번히 발생하는 최근접 질의와 영역질의를 효과적으로 수행하기 위해서 다음과 같은 연구를 수행하였다.

- 2차원 위치 정보와 각 위치에 있는 개체의 종류를 효과적으로 검색하기 위해서 인덱스를 생성하는 방안에 대해서 연구하였다. 본 논문에서는 2차원 위치 정보를 Z-ordering 방법을 사용하여 선형화 한 뒤 개체의 종류와 결합 인덱스를 생성하는 방안을 제시.
- 결합 인덱스를 B-tree를 이용하여 저장하고 Z-ordering의 특성과 영역을 둘러싸는 최소의 정사각형을 이용한 효과적인 영역질의 방법을 제안.
- 제안한 영역질의 방법을 적용한 최근접 질의 방법 제안.
- 제안한 방법을 각 지점간의 거리가 직선 거리가 아닌 실제 도로 상황으로 확장.

본 논문에서 제안한 방법은 시험적인 무인 비상 대책 시스템을 구현하는 과정에서 발생된 문제들 중에서 위치 관련 연산들을 신속히 처리하기 위해서 만들어진 방법이다. 무인 비상대책 시스템의 검색 대상들이 희소 공간에 존재하게 되는 환경을 적절히 이용한 방법으로 평가할 수 있다. 하지만 실제 도로 상의 여러 가지 정보들이 빨리 변화하는 것과 계산 대상이 많은 것을 고려한다면 현재 사용하는 디스크 상주형 시스템 보다 주기억장치 상주형 시스템을 고려하는 것도 좋은 연구 과제가 될 것으로 평가된다.

참 고 문 헌

[1] 박기석, 편저 "GIS 지리정보시스템", 동서, 1995.

[2] 최종욱, "첨단 교통정보 시스템(ATIS : Advanced transportation Information System)", 정보과학회지, Vol.109, pp.14-22, 1998.

[3] R. Bliujute, C. S. Jensen, S, Saltenis, and G. Slivinskas, "R-tree Based Indexing of now-relative bitemporal data," In the Proc. of the 24th VLDB Conf, pp.345-356, 1998.

[4] Don Chamberlin, "A Complete Guide To DB2 Universal Database," Morgan Kaufmann, 1998.

[5] Stefano Ceri and Piero Fraternali, "Designing Database Applications with Objects and Rules," Addison-Wesley, 1997.

[6] Christos Faloutsos, "Searching Multimeia Databases By Content," Kluwer Academic Publishers, 1996.

[7] Antonin Guttman, "R-trees : a dynamic index structure for spatial searching," In the Proc. of ACM SIGMOD Conf, 1984.

[8] Graham Hamilton, and Rick Cattell, "JDBC : A Java SQL API," Sun Micro systems, 1997.
[9] Graham Hamilton, Rick Cattell, and Maydene Fisher, "JDBC Database Access with Java," Addison Wesley, 1997.
[10] E. Horowitz, S. Shani, and Anderson-Freed, "Fundamentals of Data Structures in C," Computer Science Press, 1993.
[11] H. V. Jagadish, "Linear Clustering of Objects with Multiple Attributes," In the Proc. of ACM SIGMOD Conf., pp.332-342, May 1990.
[12] G. Kollios, D. Gunopulos, and V. J. Tsotras, "On indexing Mobile Objects," In the Proc. of ACM PODS Conf. pp. 261-272, 1999.
[13] Jack A. Orenstein, "Redundancy in spatial database," In the Proc. of ACM SIGMOD Conf. pp.294-305, 1985.
[14] Jack A. Orenstein, "Spatial query processing in an object oriented database system," In the Proc. ACM SIGMOD Conf., pp.326-336, 1986.
[15] Jack A. Orenstein, "A Comparison of Spatial Query Processing Techniques for Native and Parameter Spaces," In the Proc. of ACM SIGMOD conf., pp.343-352, 1990.
[16] Nick Roussopoulos, Stephen Kelly, and Frederic Vincent, "Nearest Neighbor Queries," In the Proc. of ACM SIGMOD Conf. pp.71-79, 1995.
[17] L. Arge, V. Samoladas, and J. S. Vitter. "On Two dimensional Indexability and Optimal Range Search Indexing," In the Proc. of ACM PODS Conf. pp.346-357, 1999.

김 정 일

e-mail : mokador@lghitachi.co.kr
1989~1997년 계명대학교 공과대학 전자계산학과(학사)
1998~2000년 계명대학교 대학원 컴퓨터공학과(석사)
2000~현재 LG 히다찌 공공개발팀
관심분야 : 공간데이터베이스, 실시간 데이터 베이스 처리, 데이터베이스 응용

홍 동 권

e-mail : dkhong@kmucc.keimyung.ac.kr
1985년 경북대학교 전자공학과 졸업(학사)
1992년 University of Florida 전자계산학과 졸업(석사)
1995년 University of Florida 전자계산학과 졸업(박사)
1985년~1990년 한국전자통신연구원
1996년~1997년 한국전자통신연구원
1997년~현재 계명대학교 컴퓨터전자공학부 전임강사
관심분야 : 능동 실시간 데이터베이스, 병렬 처리, 성능 평가, 시뮬레이션, 멀티미디어 처리