# 위치기반 경보서비스를 위한 MDR위치조회 알고리듬

## A MDR Location Polling Algorithm for Location Based Alert Service

**안병익*** / Byung-Ik Ahn
**양성봉**** / Sung-Bong Yang

**요 약**

무선인터넷 기술의 발달과 응용의 확산으로 위치정보를 이용한 위치기반서비스 형태는 더욱더 다양해 지고 있다. 위치기반 서비스 중 특정지역에 진입하거나 벗어날 경우에 자동으로 정보를 제공하는 위치기반 경보서비스는 향후 LBS의 가장 중요한 서비스가 될 것으로 전망된다. 그러나 위치기반 경보서비스를 제공하기 위해서는 대상이 되는 이동체의 위치정보를 주기적으로 빈번하게 조회해야 한다. 이러한 방법은 이동성을 가지는 수많은 이동체의 위치정보를 지속적이면서도 대량으로 조회해야 함으로써 심각한 시스템의 부하와 비용증가를 초래한다. 지금까지 연구되고 있는 이동체를 위한 위치조회 방법은 위치기반 경보서비스를 제공 하기 위한 효율적인 위치조회 및 구조에는 적합하지 않다. 이에 본 논문에서는 위치기반 경보서비스를 제공하기 위해 대용량의 위치정보를 조회함에 있어서, 이동체의 이동 패턴을 이용하여 불필요한 위치정보 조회 회수를 줄임으로써 시스템의 부하를 줄이는 효율적인 이동체 조회 방법을 제시하고 기존의 방법과 비교 실험하고자 한다.

**Abstract**

Location-Based Services(LBS) has been varied and expanded rapidly in local and overseas markets due to technology developments and expanded applications of wireless internet. Location Based Alert Service(LBA) capable of automatically furnishing data when entering or outing a specific location is expected to become one of the most important services in LBS. For LBA operation, it is essential to periodically get location information about moving object. However, this can cause a serious system load because system should continuously and largely receive location information of many moving objects. Existing and current methods for location polling of moving object are not suitable for an efficient location acquisition and a search structure required for LBA. In this study, to acquire large-scaled location information for LBA, a MDR moving object location polling algorithm will be suggested to reduce unnecessary location information and decrease system load by using mobility patterns of moving object.

**주요어 :** 위치기반서비스, 이동객체, 경보서비스, 이동객체 데이터베이스
**Keyword :** LBS, Moving Object, Alert Services, MODB

■ 논문접수 : 2006.10.13 ■ 심사완료 : 2006.12.21

* 교신저자 포인트아이 주식회사 대표이사 (biahn@pointi.com)
** 연세대학교 컴퓨터과학과 교수 (yang@cs.yonsei.ac.kr)

## 1. 서 론

최근 들어, 통신 및 네트워크 기술의 발달에 따라 무선인터넷을 이용한 다양한 종류의 위치기반서비스(LBS: Location Based Service)가 제공되고 있다. 위치기반 서비스는 이동 통신망을 이용하여 사람이나 사물의 위치를 실시간으로 파악하고 이를 활용한 다양한 응용 시스템 및 서비스를 통칭한다. 위치기반 서비스는 주변 시설 정보, 위치 추적, 교통, 항법, 전자상거래 등 다양한 형태의 서비스 제공이 가능하여 점차 사용자가 증가하고 있는 무선인터넷 분야의 핵심 응용 서비스이다[1,2,3].

위치기반 경보 서비스(Location-based Alert Service)는 이동통신 네트워크상에서 단말기 사용자의 위치를 모니터링하여 특정한 지역이나 설정된 영역에 진입(entering), 존재(being), 퇴거(outing)등의 이벤트가 발생될 경우 사용자에게 SMS등을 통하여 알리거나, 사용자에 의해 미리 정의된 특정한 서비스를 제공하는 서비스다. 위치기반 경보 서비스는 일종의 push형 서비스에 속하며, 매우 개인화된 서비스이다. 이런 서비스의 예로는, 위치 기반 광고 서비스, L-Commerce, 위치기반 만남/매칭 서비스, 오염지역 경보 서비스, 재난재해감지 서비스, 물류관제 서비스 등이 있다[4,5,6]. 위치기반 경보서비스의 대상 이동체가 증가하면 증가할수록 시스템의 통신부하에 의한 성능은 심각하게 저하 될 수 있으며, 이동체를 모니터링 하기 위한 비용도 크게 증가할 것이다. 이동체를 모니터링 하는 과정에서 이동체의 개수가 수백만 이상이 될 경우 위치 조회 및 이동체를 모니터링 하기 위한 이동통신 네트워크의 부하는 큰 폭으로 증가하게 될 것이다. 예를 들어 대상 이동체가 300만개이고 위치 조회 시간 간격이 1분이라고 하였을 경우 하루 동안의 총 위치 조회 회수는 43억2천만(3,000,000 * 1440)번으로 위치기반 경보서비스를 이동통신 네트워크 상에서 적용할 수가 없다. 따라서 이동체를 모니터링 하는 과정에서 같은 경보조건에 대하여 동등한 경보 서비스가 발생 되도록 하면서도 이동체에 대한 위치 조회 횟수를 줄이는 것이 위치기반 경보서비스에서 아주 중요한 문제이다[7,8,9].

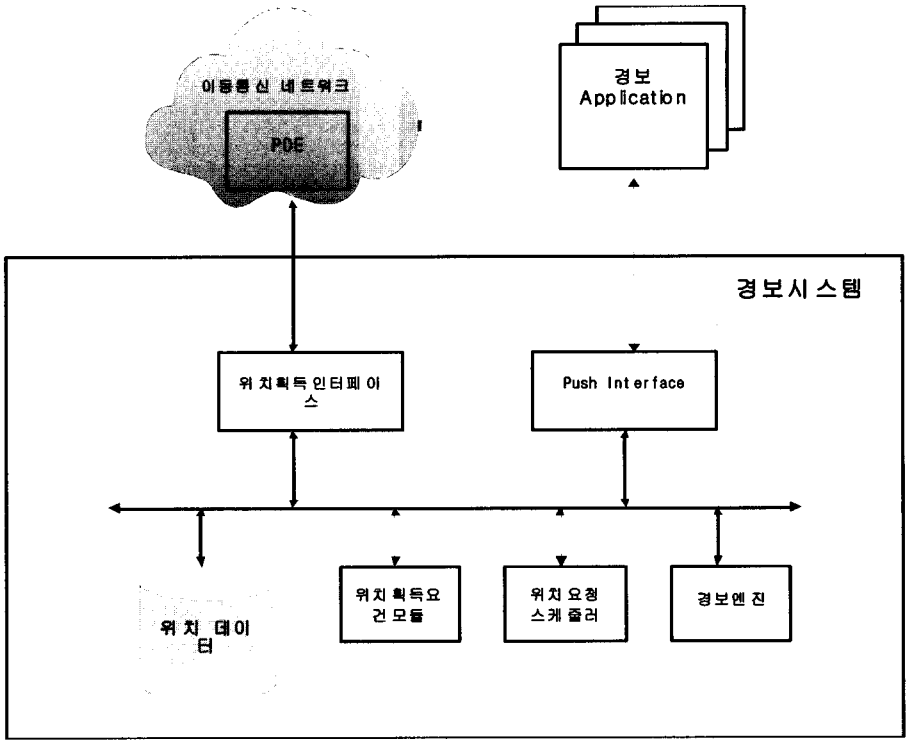
본 논문에서는 이동체의 이동 패턴을 파악하여 조회 회수를 줄임으로써 통신부하를 줄이기 위한 효율적인 이동체 조회 방법을 제시하고 기존의 방법과 비교 실험을 하였다.

## 2. 위치기반 경보서비스 시스템

이동체를 모니터링 하는 과정에서 같은 경보조건에 대하여 동등한 경보 서비스가 발생되도록 하면서도 이동체에 대한 위치 조회 횟수를 줄이는 것이 위치기반 경보서비스에서 아주 중요한 문제이다.

이러한 위치기반 경보서비스의 중요 기술은 위치발생 장치가 어떤 방식인지에 대한 것과 위치조회를 어떤 스케줄링에 의하여 하는가에 관한 것이 있다. 위치발생장치의 종류와 관련하여는 위치 측위 발생 장치가 서버인지, 단말인지가 문제가 되고, 위치조회를 위한 스케줄링에서는 어떤 방법으로 이동 단말의 위치를 주기적 혹은 계산에 의해 설정된 시간 안에 측위 하느냐 하는 것이 문제가 된다. 일반적으로 위치발생 장치는 각각의 방식에 상관없이 일정 주기, 예를 들면 5분마다 서버 혹은 단말에서 위치 측위를 하여 트리거 조건을 검사하여 보고한다. 그러나 이동통신과 같은 다수의 사용자를 갖는 시스템에서는 너무 많은 위치 측위를 발생시켜 네트워크 부하가 커질 수 밖에 없다[10].

<그림 1>은 무선 통신 네트워크에서 이용하기 위한 경보서비스 시스템 구성도이다. 경보시스템은 네트워크에 대한 위치 조회 인터페이스를 갖는다. 이 인터페이스는 네트워크 내의 개별적인 이동체들에 갱신된 위치 데이터 요구들을 전송하고 이 개별적인 이동체들에 대한




<그림 1> 경보서비스 시스템 구성

위치 데이터를 수신하는 데에 이용된다[8,11]. 경보시스템는 또한 Push 인터페이스를 통해 1개 이상의 경보 기반 애플리케이션들과 통신한다. Push 인터페이스는 경보 기반 애플리케이션들로부터 경보 트리거링 조건들 및 이동체 아이덴티피케이션을 수신하고, 위치 기반 경보 서비스에 대응하는 경보 트리거링 조건들이 네트워크 내의 특정한 이동체에 의해 충족되는 때를 나타내는 정보를 각 경보 기반 애플리케이션에 전송하는 데에 이용된다.

위치 데이터 DB는 다수의 데이터 요소들을 포함하는데, 각 데이터 요소는 이동체의 아이덴티티, 이동체의 위치 및 PDE에 의해 위치가 결정되었던 시간과 관련 있다. 경보시스템은 위치조회 요건엔진, 위치요청 스케쥴러 및 경보 엔진을 포함한다. 위치조회 요건 엔진은, 특정한 이동체에 대해 언제 위치 갱신이 이루어지는 지의 지식을 얻기 위해 위치데이터 DB와 인터페이스한다. 위치조회 요건엔진은 또한 개별적인 이동체들에 대해 위치 기반 경보 서비스에 대응하는 경보 트리거링 조건들에 관한 정보를 얻기 위해, Push 인터페이스를 통해 경보 기반 애플리케이션과 인터페이스한다. 위치조회 요건엔진은 위치요청 스케쥴러에 메세지를 출력하는데, 이 메세지는 특정한 이동체에 대한 갱신된 위치 데이터가 요구되는 데드라인(deadline)을 나타내는 데이터 요소를 포함한다.

위치요청 스케쥴러는 위치조회 요건 엔진에 의해 출력된 메세지를 수신한다. 이 요청 스케쥴러는 위치 요구들이 스케쥴링되는 때를 결정하기 위해 위치조회 인터페이스를 통해 상기 네트워크와 인터페이스한다. 요청 스케쥴러는 적절한 컴퓨팅 플랫폼 상에서 실행되는 소프트웨어로 구현된다. 특정한 이동체 및 데드라인을 식별하는 위치조회 요건 엔진으로부터 수신된 메세지 내의 데이터 요소에 기초하여, 위치

요청 스케쥴러의 역할은 특정한 이동체에 대한 갱신된 위치 데이터가 상기 데드라인 전에 위치 데이터 DB에 기록되도록 보장하는 것이다.

경보 엔진은 특정한 이동체에 대한 갱신된 위치 데이터가 언제 수신되는 지의 지식을 얻기 위해 위치 데이터 DB와 인터페이스한다. 경보 엔진은 특정한 이동체에 관련된 위치 기반 경보 서비스들에 대응하는 경보 트리거링 기준들을 얻기 위해 경보 기반 애플리케이션들과 인터페이스한다. 위치 데이터 DB로부터 얻어진 정보에 기초하여, 경보 엔진은 특정한 이동체가 위치 기반 경보 서비스에 대응하는 경보 트리거링 조건들을 충족하는 지를 결정한 다음, 조건들을 충족한다면, 경보를 발생시켜야 함을 나타내는 트리거를 경보 기반 애플리케이션에 전송한다.

위치기반 경보서비스를 제공하기 위한 위치 조회 방법은 크게 서버 위치 조회 과 단말 위치 조회로 구분된다. 서버 위치 조회는 위치 조회 요청의 주체가 서버인 경우이고 단말 위치 조회는 위치조회 요청의 주체가 단말인 경우에 해당된다.

## 3. MATT 최소경보 트리거 시간 위치 조회 방법

웨이브얼럿(WaveAlert) 방법은 최소경보 트리거시간(MATT : Minimum Alert Triggering Time)과 최근 가능 갱신 시간(EAUT : Earliest Available Update Time)을 계산하여 위치조회 시간을 조절하는 위치기반 경보시스템이다[12]. MATT는 단말기 사용자의 위치에서 목적 위치까지의 계산된 거리(유클리드 거리, Shortest Path 거리 등)를 이동체의 최대 가능 속도로 나눈 시간 값으로 경보 조건을 충족하는 가장 빠른 미래시간이다. 이는 위치데이터를 수신하기 위한 데드라인으로 정의될 수 있다. EAUT는 요청을 통해 조회된 데이터가 새로운 MATT을 갱신할 수 있는 가장 빠른 수용 가능한 갱신 시간으로 이 이전의 조회된 위치 데이터는 무시된다. 즉, EAUT와 MATT 시간 사이에 측위된 위치데이터만이 유효하며, 다음 MATT 혹은 EAUT를 갱신하는데 이용된다[12].

위치 갱신들이 요구되는 빈도수를 최소화하기 위한 요건 엔진의 목적은 특정한 이동체에 대한 갱신된 위치 데이터를 수신함에 있어서, 수용가능 한 지연을 허용함으로써 달성된다. 그러나, 이러한 지연은 위치기반 경보서비스들에 대응하는 경보 트리거링 조건들을 놓치지 않음을 보장할 수 있도록 해야 한다. 따라서, 특정한 이동체 및 특정한 경보 트리거링 조건들과 관련하여, 요건엔진은 특정한 이동체에 의해 경보 트리거링 조건이 충족될 수 있는 기대되는 가장 빠른 미래의 시간을 결정한다. 이는 특정한 이동체 및 특정한 경보 트리거링 조건에 대한 최소 경보 트리거 시간(MATT)으로서 칭해질 수 있다.

일단 계산되면, MATT는 특정한 이동체에 대한 갱신된 위치 데이터가 특정 시간 전에 수신되는 요건으로 공식화되며, 이러한 요건은 요구 스케쥴러에 전송되는 메세지 내에 데이터 요소로서 공급된다. 이러한 방식으로 결정되는 시간 인스턴트는, 이동체가 MATT 이전에는 특정한 경보 트리거링 조건을 만족시키지 못할 것이며, 이에 따라 그 사이의 어느 시간에 수신된 갱신된 위치 데이터는 대응하는 위치 기반 경보 서비스를 발생시키지 않을 것이라는, '경험에서 나온 추측(educated guess)'을 나타낸다. MATT는 또한 갱신된 위치 데이터를 수신하기 위한 '데드라인' 또는 만료 시간으로서 칭해질 수 있다[9].

## 4. MDR 이동 비율 기반 위치 조회 방법

위치기반의 경보 서비스를 하기 위해서는 대상 이동 단말 사용자가 설정한 임의의 경보 영역 내에 위치할 때까지 지속적인 위치 조회가 필요하다. 쉽게 고려할 수 있는 방법으로 일정 시간 간격으로 대상 이동 단말에 대한 위

```
Δt : time interval
Dmin : minimum moving distance, Dmax : maximum moving distance
Tinitial : initial time interval
PollingTime : a set of Timer
TimeInterval : Δt ∈ { Δt× 1/n, ···, Δt× 1/4 , Δt× 1/3 , Δt× 1/2 , Δt ,
                Δt× 2 , Δt× 3 , ···, Δt× n }
① for all each time interval Δt in TimeInterval,
   PollingTime[interval] = new Timer(Tinitial) // timer instance
② for all each moving object mObject,
   PollingTime[0].Queue(mObject)
③ for all each Timer in PollingTime,
   Timer.Start( ) // location polling start
④ PollingTime[interval].Callback( )
   // pollingTime[interval] 시간이 되면 호출되는 Callback 함수
  ④-1 for all moving object mObject in pollingTime[interval].Queue
       call location.polling(mObject)
⑤ location.polling(mObject)
  ⑤-1 mObject.Dbefore = mObject.distance // 이전 이동거리 저장
  ⑤-2 update mObject.x, mObject.y, mObject.time // 위치갱신
  ⑤-2 update mObject.distance // 현재 이동거리 갱신
  ⑤-3 MDR.Reset(mObject) // 위치갱신 간격 조절
  ⑤-4 MDR.Alert(mObject) // 위치 경보 처리
  ⑤-5 interval = TimerQueue(mObject.Δt)
  ⑤-6 this mObject, pollingTime[interval].Queue(mObject)
⑥ MDR.Reset(mObject)
  ⑥-1 α = mObject.distance/mObject.Dbefore // 이동비율 계산
  ⑥-2 if mObject.distance <= Dmin then
           increase mObject.Δt, increased Δt ∈ TimeInterval
      else if mObject.distance >= Dmax then
           decrease mObject.Δt, decreased Δt ∈ TimeInterval
      else if α <= 1 then
           decrease mObject.Δt, increased Δt ∈ TimeInterval
      else decrease mObject.Δt, in``creased Δt ∈ TimeInterval
⑦ MDR.Alert(mObject)
  ⑦-1 if mObject.bufferZone not contain mObcet then
     return this function
      else if mObject.alertZone contain mObject then
           LocationAlertEvent() // Alert Event 발생
      else mObject.Δt = Tinitial // buffer zone 에 있는 경우□
```

<그림 2> LBA를 위한 MDR 위치조회 알고리즘

치 요청을 할 수 있다. 하지만 다수의 사용자에 대해 일정 시간 간격으로 위치 요청을 하는 것은 수많은 위치 요청의 발생을 초래한다. 이에 일정 시간 간격이 아닌, 경보 서비스를 하기에 적합하며 위치 조회 횟수를 최소화 할 수 있는 방법이 적용되어야 한다. 이동 비율기반 위치 조회 방법은 이동체의 움직임을 관찰하면서 위치 조회 시간간격을 적절하게 조절 해 줌으로써 위치조회 횟수를 최소화 하면서 정확한 경보서비스 제공이 가능하다.

이동 거리 비율기반 위치조회에 사용되는 인자들 및 자료구조는 다음과 같이 정의 및 처리 할 수 있다.

(i) Tinitial은 초기의 시간 간격이다.

(ii) mObject.distance는 현재 위치의 이동거리 이고, mObject.Dbefore는 이전 위치의 이

동 거리이다.

(iii) $\Delta t$는 다음 위치 요청을 위한 시간 간격으로서 현재의 시간인 tcurrent 시점에 위치 요청을 한 후 tcurrent + $\Delta t$의 시점에 다음 위치 요청을 하게 된다.

(iv) 현재 시간(tcurrent)과 이전 시간(tbefore)의 이동 거리 비율인 $\alpha$는 mObject.distance/mObject.Dbefore 로 정의되며, 이는 이전의 이동 거리에 비해 현재 얼마만큼 더 이동 하였는지를 나타낸다.

(v) Dmin은 이동이 거의 없다고 간주할 수 있는 최소 이동 거리이며, Dmax는 이동 거리의 폭이 상당히 크다고 간주할 수 있는 최대 이동 거리이다.

이전의 이동 거리와 현재의 이동 거리의 비율에 따라 다음 위치 조회를 하기 위한 시간 간격($\Delta t$)을 증가, 감소 시킴으로써 위치 요청 횟수를 최소화하기 위한 알고리즘은 <그림 2>와 같다.

PollingTime는 여러 개의 시간 인스턴스를 가지고 thread로 동작하며, 주어진 시간이 경과하면 자동적으로 CallBack 함수를 호출하게 하는 시간 인스턴스의 집합이다. 고정간격으로 위치를 조회하는 정적 (Static) 알고즘에서는 모든 이동체의 위치 조회 간격이 동일 하기 때문에 Tinitial로 설정된 시간 인스턴스 만이 필요하였지만 MDR에서는 이동체 각각 다른 시간 인스턴스들을 갖는다. 모든 이동체에 대하여 시간 큐에 삽입을 하고 위치조회 구동을 시작한다. 큐에 들어 있는 이동체들은 초기 시간 간격인 Tinitial 이 지난 후에 CallBack 함수로 호출이 되고, 각각의 이동체들은 Location.Polling 함수를 통하여 위치를 조회 한다. Location.Polling에서는 먼저 위치를 갱신 하고 위치조회 시간 간격을 재설정하는 MDR.Reset을 호출한다. MDR 알고리즘에서는 $\Delta t$는 정해진 시간 간격 또는 특정한 계산식을 가지고 사용될 수 있다. 다음 위치조회를 위한 $\Delta t$ 를 설정한 후에는 위치 경보 처리 함수 인 MDR.Alert를 호출하며, 이 함수는 해당 이동체가 경보조건을 만족하는 경보영역에 진입했는지를 체크하여 처리 한다. 상기 과정을 마친 후에 다음 위치조회를 위한 시간 큐에다가 이동체를 넣기 위하여, 시간 인스턴스 생성을 위한 'interval' 값을 TimerQueue 함수를 통하여 얻는다. 여기서 리턴 된 값을 가지고 시간 큐에다가 이동체를 삽입한다. 큐에 들어 있는 이동체들은 이제 각각 생성된 시간 간격에 따라 일정한 시간이 되면 CallBack 함수를 통하여 호출되어 상기의 과정을 반복 한다.

## 5. LBA를 위한 위치조회 알고리즘 실험

본 장에서는 위치기반 경보서비스를 위한 이동비율기반 위치조회 알고리즘에 대하여 비교 성능 실험을 하기 위하여, 이동체 데이터 실험에 대표적으로 사용되는 도구인 GSTD(Generation of SpatioTemporal Datasets) 생성기[13]를 사용하였다. 본 논문에서 제시하는 이동비율기반 위치조회 알고리즘인 MDR 과 최소경보트리거 시간 알고리즘인 MATT 그리고 정적으로 위치조회를 수행하는 방법에 대하여 테스트 환경을 구성하여 각각 비교 실험하였다. 경보서비스를 위한 위치조회 방법들은 JAVA로 구현하였으며, AMD 64비트 1.6GHz 프로세서, 1GB 메모리와 Windows XP를 사용하여 실험을 수행하였다.

이동체 위치 데이터를 실험하기 위한 요소는 <표 1>과 같이 이동체 개수는 1000개를 대상으로 실험을 하였으며, 실험시간 time stamp는 1 unit을 5초로 하여 총 10000개의 time stamp를 갖도록 하였다. 따라서 총 실험 시간 영역은 14시간이고 실험 대상공간은 100Km * 100Km로 설정하였다. 실험 공간을 100Km * 100Km로 설정한 이유는 안전, 광고 등 대다수의 위치기반 경보서비스가 가장 많이 쓰이는 공간 영역이 대도시를 중심으로 주변 영역을 포함한 공간이기 때문에 그 크기가 100Km * 100Km를

<표 1> GSTD 데이터의 실험요소

| 실험 환경 요소 | 내 용 | 비 고 |
|---|---|---|
| 이동체 개수 | 1,000 | |
| 실험 시간 Time Stamp | 10,000 | 1unit = 5 sec |
| 총 실험 시간 | 14시간 | 10,000 * 5 sec = 13.88 시간 |
| 이동체의 이동패턴 | Uniform, Gaussian, Skewed | |
| 실험공간 이탈여부 | No | |
| 실험 공간 크기 | 100Km * 100Km | |
| 이동체 경보영역수 | 15~20개 | |
| 경보영역 크기 | 1~5Km | |

넘지 않는 것이 가장 이상적이다. 또한 실험공간 이탈 여부는 이동체가 실험공간을 벗어나는 경우에 이탈 처리를 하는 경우와 이탈하지 않고 다시 방향을 바꾸어 영역 안으로 들어오게 하는 방법 두 가지가 있는데, 본 실험에서는 이탈처리를 하지 않는 것으로 설정하였다.

이동체의 이동패턴은 Uniform, Gaussian, Skewed 등 3가지 분포를 가지고 실험을 하였다. Uniform 위치 데이타 분포는 위치 데이타가 실험 공간에서 균등하게 분포된 형태를 의미한다. Gaussian 위치 데이터 분포는 위치 데이터가 정규분포를 가지는 형태를 의미 한다. Skewed 위치 데이타 분포는 위치 데이터가 비대칭 분포를 가지는 형태를 의미 한다. 1개의 이동체 마다 가질 수 있는 경보 영역 수는 15개~20개로 실험을 하였다. 즉, 이동체 하나 당 Alert를 발생시키는 경보영역이 15개 이상 20개 이하이며, 이동체의 위치가 업데이트 될 때마다 경보 영역에 진입했는지를 파악하여 Alert를 발생 시키게 된다. 또한 경보 영역의 크기는 1~5 Km로 설정하였다.

<표 2>는 위치조회 방법을 실험하기 위한 인자 값으로 MDR(이동거리비율기반 위치조회 알고리즘)에서 사용하는 버퍼영역은 1~6 Km의 값으로 실험을 하였으며, 위치 조회 시간 간격 최소치인 T_MIN는 5분으로, 최대 한계치인 T_MAX는 60분으로 설정하였고 최초의 시작시간 간격인 T_INIT는 5분으로 설정 하였다. MATT에서 사용하는 이동체의 속도 값은 20~120 Km를 가지고 실험을 하였다.

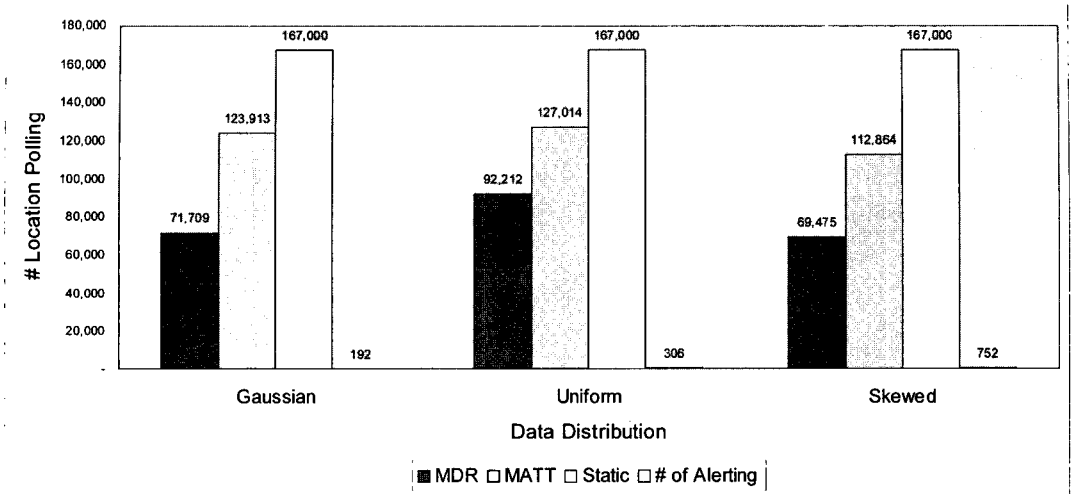
<표 2> 위치조회 알고리즘 인자

| 알고리즘 인자 | 값 | 비고 |
|---|---|---|
| 버퍼영역 | 1~6 km | MDR버퍼영역 |
| T_INIT, T_MIN | 5분 | 최소한계치 |
| T_MAX | 60분 | 최대한계치 |
| V-MAX | 20~120 km/h | 최고속도 |

또한 본 논문에서 위치기반 경보서비스를 위해 제시하는 MDR 알고리즘을 비교 실험하기 위하여 같은 조건하에서 정적(Static) 알고리즘과 MATT 알고리즘을 구현하여 비교실험을 하였다. 이 3가지 알고리즘은 기본적인 시간간격 생성 방법과 경보처리 방법, 데이터 구조 등을 동일하게 구성하여 위치기반 경보서비스를 위한 각각의 효율성을 투명하게 비교 할 수 있도록 하였다.

### 5.1 데이터 분포에 의한 위치조회 알고리즘 실험

<그림 3>은 데이터 분포에 따른 MDR, MATT, Static 알고리즘의 실험 결과 이다. 각 분포 별로 이동체가 경보영역에 진입하여 alert가 발생한 횟수는 Skewed분포가 752건으로 제일 많았고 그 다음이 Uniform 분포가 306건, Gaussian 분포가 192건으로 제일 적었다. Skewed 분포가 제일 많았던 이유는 데이터가 한쪽으로 편향되어 있기 때문에 경보 영역에 들어갈 확률이 높은 것으로 분석된다. Static 알고리즘의 경우는 분포에 상관없이 167,000건으로 동일하게 위치 조회를 하였다. 이는 고정 시간 간격인 5분단위로 위치조회를 하였기 때문이다.




<그림 3> 데이터 분포에 따른 MDR/MATT/Static알고리즘 실험

MATT의 경우 Uniform 분포가 127,014건, Gaussian 분포가 123,913건, Skewed 분포는 112,864건으로 결과가 나왔다. 따라서 MATT는 Static 알고리즘 보다 약 Uniform 분포에서는 24%, Gaussian 분포에서는 23%, Skewed 분포에서는 33%의 위치조회 횟수를 감소하는 효율을 보여주고 있다. MDR 알고리즘의 경우는 Uniform 분포가 92,212건, Gaussian 분포가 71,709건, Skewed 분포는 69,475건으로 결과가 나왔다. 따라서 MDR알고리즘은 Static 알고리즘 보다 약 Uniform 분포에서는 45%, Gaussian 분포에서는 58%, Skewed 분포에서는 59%의 위치조회 횟수를 감소하는 효율을 보여주고 있다. 두 알고리즘 다 Skewed 분포에서 효율이 높고 그 다음이 Gaussian 분포, 제일 낮은 것이 Uniform 분포이다. Skewed 분포에서 효율이 높은 이유는 이동체들이 편향되어 있기 때문이다. 이동체들이 집중되어 있거나 편향되어 있을 경우 고르게 잘 분포되어 있는 것보다 더 적은 수의 위치 조회로 alert가 가능하다.

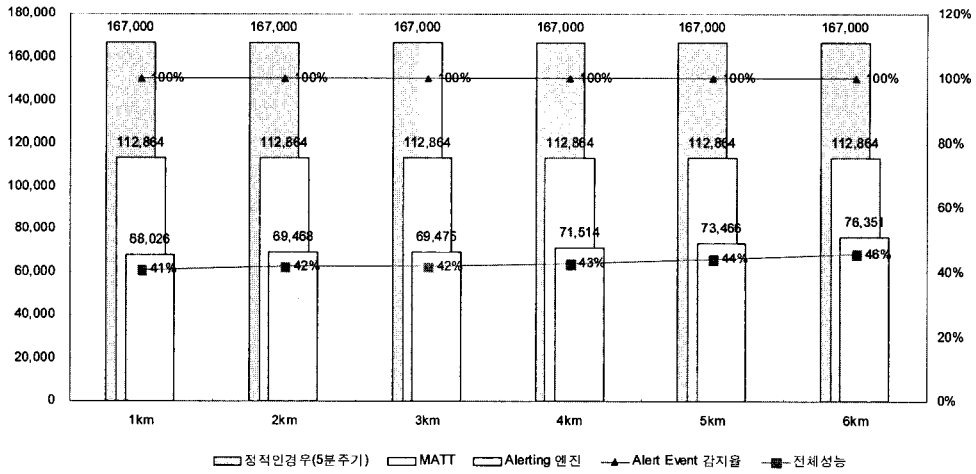
이 실험을 통하여 MDR 알고리즘이 Static 알고리즘과 MATT 알고리즘 보다 성능이 우수함을 알 수 있다. MDR 알고리즘은 Uniform 분포에서 MATT 보다는 약 28%, Static 보다는 약 45%의 성능향상을 보여주고 있다. Skewed 분포에서는 MATT 보다는 약 38%, Static 보다는 약 59%의 성능향상을 보여주고 있다.

### 5.2 버퍼영역 크기에 의한 MDR 알고리즘 실험

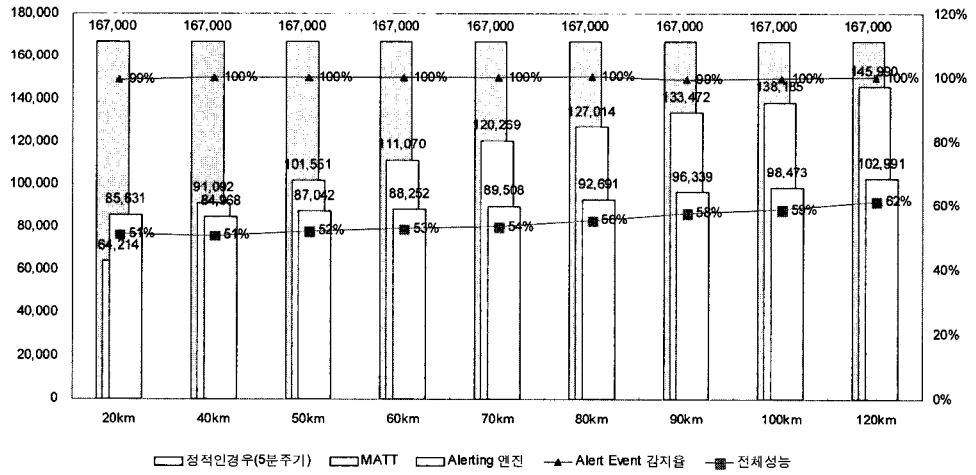
<그림 4>는 버퍼영역 크기 변경에 따른 MDR 알고리즘의 Skewed 분포에서의 성능 실험 결과이다. 버퍼영역을 1~6Km까지 변경하면서 실험을 하였다. Uniform 분포의 경우 버퍼영역이 적을수록 위치를 조회하는 횟수가 적고 버퍼영역이 클수록 위치를 조회하는 횟수가 점점 많아졌지만 Skewed 분포에서는 버퍼영역에 따른 위치조회 횟수의 차이는 그다지 크지 않았다. Skewed 분포의 경우, MDR 알고리즘은 버퍼영역에 상관없이 모두 752건의 Alert Event를 발생하였다.

### 5.3 최고속도 값 변경에 의한 MATT 알고리즘 실험

MATT 알고리즘은 최대 이동 속도 값인




<그림 4> 버퍼 영역 크기에 따른 MDR 실험 - Skewed

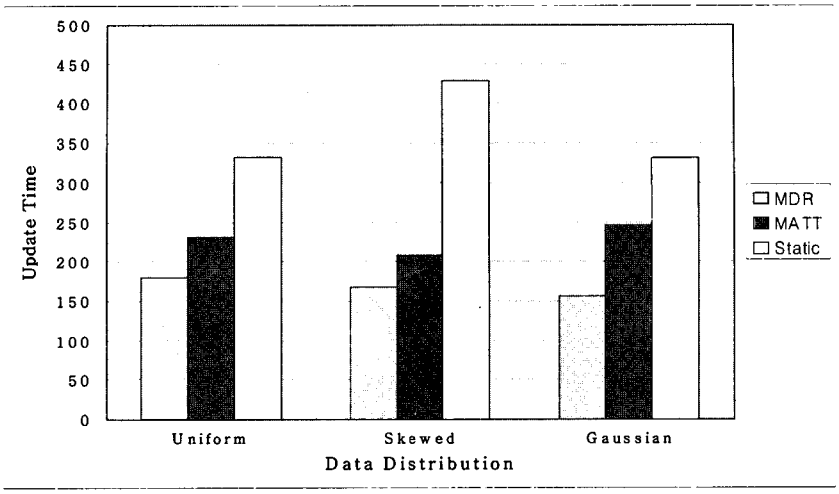



<그림 5> VMax 값 변경에 따른 MATT 실험 - Uniform

Vmax가 성능 평가에 중요한 요소이다. <그림 5>는 MATT 알고리즘을 Uniform 데이터 분포에서 Vmax 값을 20~120km까지 변경하면서 성능을 측정한 결과이다. MATT의 경우 이동속도가 더 적을수록 위치를 조회하는 횟수가 적고 이동 속도가 클수록 위치를 조회하는 횟수가 점점 많아진다. Uniform 분포의 경우, MATT 알고리즘은 최대 가능 속도가 20km 일 때 65,631건 위치조회를 하였고. 최대속도가 120Km 일 경우에는 145,990건으로 위치조회 횟수가 발생하였다. 즉, 최대 이동속도가 커질수록 위치 조회 효율은 떨어짐을 알 수 있다.

## 6. 인덱싱 적용 위치조회 알고리즘 실험

이동체는 시간에 따라 연속적으로 위치 데




<그림 6> 위치조회 알고리즘 데이터 분포 별 R*-Tree 인덱싱 갱신 시간

이터가 갱신되는 객체이다. 이런 이동체를 검색하고 질의 처리를 하기 위해서는 이동체의 위치 데이타를 이동체 데이터베이스에 저장해야 하며, 효과적인 검색 및 질의 처리를 위해 이동체 색인이 필요하다. 지금까지 다양한 이동체 색인에 대한 연구가 진행되고 있으며, 현재의 위치 및 미래 위치 색인을 위해서는 통상적으로 기존의 공간 인덱싱 기법인 R-tree 계열이 많이 사용된다.

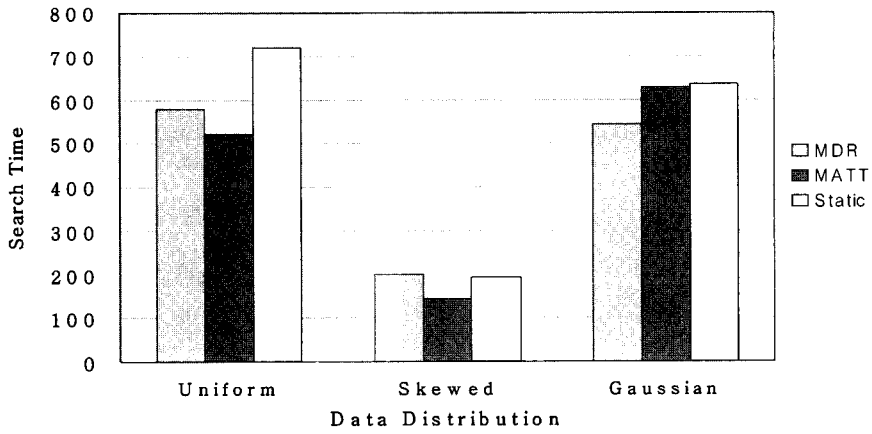
위치기반 경보서비스를 위한 위치조회 알고리즘은 현재의 위치가 경보 영역에 진입 또는 이탈 했는지를 관리하기 때문에 그 특성상 과거의 위치정보나 궤적 정보가 필요 없고 현재의 위치만 관리 하면 된다. 따라서 위치기반 경보서비스를 위한 이동체 색인 방법으로는 현재 위치 색인 및 미래 위치 색인 방법이 적합하다. 이런 이동체의 가장 큰 특징은 이동체들의 갱신이 아주 빈번하다는 것으로 색인 기법을 적용 시에 사장 영역(dead space)이 크게 발생하고, 색인을 위한 MBB(Minimum Bounding Box) 빈번한 중복으로 인하여 갱신 비용이 크게 발생한다는 단점이 있다.

이런 문제점을 해소 하기 위하여 현재 위치 색인 및 미래 위치 색인을 위해 많이 사용하는 색인 구조는 R*-Tree와 LUR-Tree(Lazy Update R-Tree)이다. 본 연구에서는 위치기반 경보서비스를 위한 제시된 MDR위치 조회 알고리즘과 이를 비교하기 위한 MATT알고리즘 및 Static 알고리즘을 각각R*-Tree와 LUR-Tree 색인으로 구현하여 실험을 하였다.

### 6.1 R*-Tree를 통한 위치조회 알고리즘 실험

R-Tree는 높이 균형 트리이기 때문에 데이터의 분포와 관계없이 기본적으로 성능이 우수하지만, 이동체의 계속되는 위치 이동으로 인하여 빈번한 색인의 변경과 이동체의 갱신이 자주 발생하여 전체적인 성능 저하를 초래 한다. R*-Tree는 R-Tree가 삽입 시에 영역만을 고려하는 단점을 보완하기 위하여 중복과 가장자리(margin)를 추가적으로 고려한 색인 구조 이다. R*-Tree는 최소 중복 증가 정책 (Least Overlap Enlargement Policy)을 사용한 삽입 노드 선택 정책과, 가장자리 값과 중복 값을 고려한 새로운 분할 기법을 갖는다. 또한 재삽입 정책을 사용하여 R-Tree의 공간 활용도가 낮은 문제점을 보완 하였다.

<그림 6>은 R*-Tree 색인 구조를 적용한



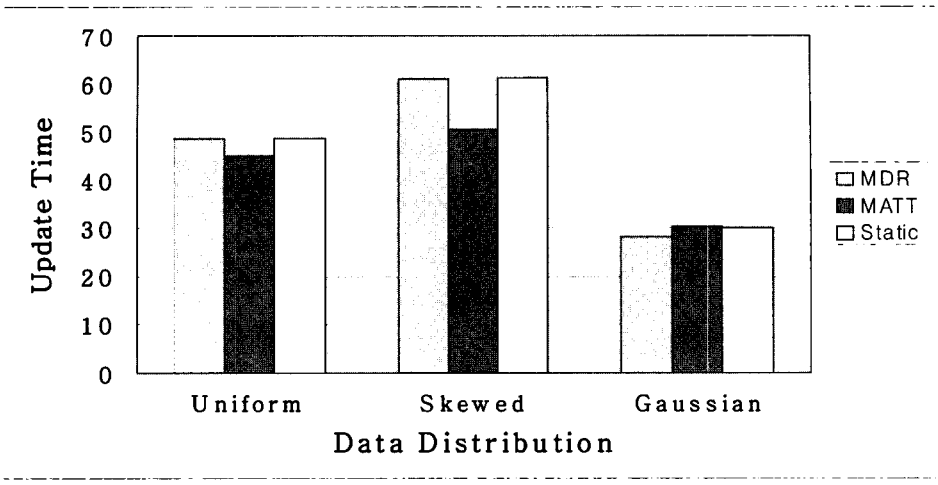

<그림 7> 위치조회 알고리즘 데이터 분포 별 R*-Tree 인덱싱 검색 시간

데이터 분포에 따른 MDR, MATT, Static 알고리즘의 이동체 위치 갱신(update) 시간 이다. 실험 결과 분포 별 갱신 시간은 Static 알고리즘의 경우 Uniform 분포가 332.465초, Gaussian 분포가 331.369초, Skewed 분포는 428.672초로 결과가 나왔다. Static 알고리즘의 경우 고정 간격으로 위치조회를 하기 때문에 3개 분포 별로 위치 조회 총수가167,000 건으로 같지만 갱신 시간은 Skewed 분포가 가장 높고, Uniform 분포와 Gaussian 분포가 비슷하게 나왔다. Skewed 분포가 갱신 시간이 가장 높은 이유는 이동체의 분포가 고르지 않고 편향되어 있기 때문에 이동체 위치 갱신 시에 색인의 변경이 좀더 발생한 것이다.

MATT의 경우 Uniform 분포가 230.566초, Gaussian 분포가 245.832초, Skewed 분포는 208.318초로 결과가 나왔다. 따라서 MATT는 Static 알고리즘 보다 약 Uniform 분포에서는 31%, Gaussian 분포에서는 26%, Skewed 분포에서는 52%의 위치갱신 시간을 감소하는 효율을 보여주고 있다. MDR 알고리즘의 경우는 Uniform 분포가 179.620초, Gaussian 분포가 157.094초, Skewed 분포는 176.666초로 결과가 나왔다. 따라서 MDR알고리즘은 Static 알고리즘 보다 약 Uniform 분포에서는 46%, Gaussian 분포에서는 53%, Skewed 분포에서는 61%의 위치 갱신 시간을 감소하는 효율을 보여주고 있다. MDR알고리즘과 MATT 알고리즘 둘 다 Skewed 분포에서 효율이 높고 그 다음이 Gaussian 분포, 제일 낮은 것이 Uniform 분포이다. Skewed 분포에서 효율이 높은(갱신 시간이 가장 적은) 이유는 이동체들이 편향되어 있기 때문이다. 이동체들이 집중되어 있거나 편향되어 있을 경우 고르게 잘 분포되어 있는 것보다 더 많은 색인의 갱신이 필요하기 때문이다.

이 실험을 통하여 MDR 알고리즘이 Static 알고리즘과 MATT 알고리즘 보다 갱신 시간 성능이 우수함을 알 수 있다. MDR 알고리즘은 Uniform 분포에서 MATT 보다는 약 23%, Skewed 분포에서는 MATT 보다는 약 20%, Gaussian 분포에서는 약 37%의 갱신 시간 성능향상을 보여주고 있다.

<그림 7>은 R*-Tree 색인 구조를 적용한 데이터 분포에 따른 MDR, MATT, Static 알고리즘의 이동체 위치 검색(search) 시간 이다. 이동체 위치 검색 시간은 이동체의 위치 갱신이 발생한 경우, 이동체와 관련된 경보영역들




<그림 8> 위치조회 알고리즘 데이터 분포 별 LUR-Tree 인덱싱 갱신 시간

에 이동체가 진입했는지를 판단하기 위하여 경보영역을 질의한 검색 시간이다.

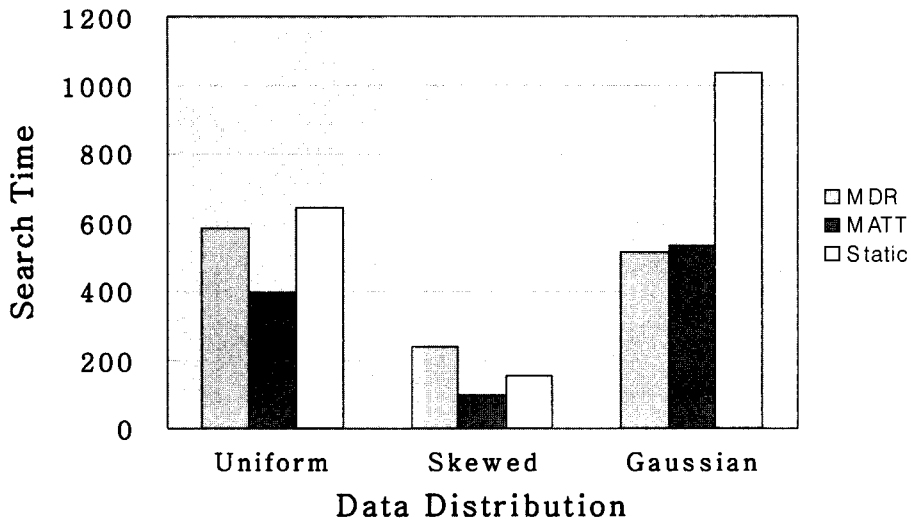
실험 결과 분포 별 갱신 시간은 Static 알고리즘의 경우 Uniform 분포가 719.323초, Gaussian 분포가 635.262초, Skewed 분포는 193.445초로 결과가 나왔다. Static 알고리즘의 경우 고정간격으로 위치조회를 하기 때문에 3개 분포별로 위치 조회 총수가167,000 건으로 동일하지만 검색 시간은 Uniform 분포가 가장 높고, 그 다음이Gaussian 분포, 그리고 Skewed 분포가 가장 적은 검색시간으로 나왔다. Uniform 분포가 검색 시간이 가장 높은 이유는 이동체의 분포가 고르게 분산되어 있기 때문에 검색을 위한 경보 영역 검색 횟수가 많아서이다.

MATT의 경우 Uniform 분포가 521.848초, Gaussian 분포가 627.953초, Skewed 분포는 143.389초로 결과가 나왔다. 따라서 MATT는 Static 알고리즘 보다 약 Uniform 분포에서는 우수한 성능을 보였고, Gaussian 분포와 Skewed 분포에서는 근소하지만 약간 적은 위치 검색 시간이 소요되었다. MDR 알고리즘의 경우는 Uniform 분포에서는 검색 시간이 577.807, Gaussian 분포가 543.337초, Skewed 분포는 198.358초로 결과가 나왔다. 따라서 MDR알고리즘은 Gaussian 분포에서는 Static 알고리즘과 MATT 알고리즘보다 성능이 우수한 결과가 나왔다. 그러나 Uniform 분포와 Skewed 분포에서는 Static 알고리즘 보다는 우수하지만, MATT 보다는 검색 시간이 약간 더 걸리는 것으로 결과가 나왔다. MDR알고리즘과 MATT 알고리즘 둘 다 Uniform 분포에서 효율이 높고 그 다음이 Gaussian 분포, 제일 낮은 것이 Skewed 분포이다.

MDR 알고리즘이 Uniform 분포와 Skewed 분포에서 MATT알고리즘 보다 검색 시간이 약간 많은 이유는 MDR 알고리즘의 버퍼영역과 관련이 있다. 이 실험의 결과는 MDR 알고리즘의 버퍼 영역을 3Km로 설정하고 한 결과이다. 버퍼 영역 크기에 따라 검색을 위한 노드 방문횟수가 달라진다. 버퍼 영역이 클수록 검색을 위한 노드 방문 횟수가 증가하고 버퍼 영역이 적을수록 검색을 위한 노드 방문 횟수가 줄어든다.

### 6.2 LUR-Tree 적용 위치조회 알고리즘 실험

LUR-Tree는 공간색인인 R-tree를 이용한 이동체 색인 방법으로 갱신 비용을 줄이는 것이 목적이다. 기존의 R-tree는 이동체의 모든 위치변경은 갱신 연산을 전부 수행해야 한다.




<그림 9> 위치조회 알고리즘 데이터분포별 LUR-Tree 인덱싱 검색 시간

갱신 연산은 노드 분할 및 합병을 유발 하기 때문에 이동체 데이터의 경우R-Tree 에서의 갱신 비용은 아주 높다. LUR-Tree는 이런 문제점을 해결하기 위하여 이동체의 위치변경으로 발생하는 갱신을 일정 기간 지연시킴으로 해서 갱신으로 인해 빈번하게 발생하는 노드 분할 및 합병 횟수를 줄이는 것이 특징이다.

LUR-Tree에서는 객체가 속해 있는 노드의 MBR내에서 객체가 이동한 경우는 MBR을 수정할 필요가 없으므로 트리의 갱신을 지연시킨다. 즉, 객체를 지우고 다시 삽입하는 과정을 수행하지 않는다. 또한 객체가 MBR을 벗어나서 움직인 경우(Lazy Update의 경우가 아닐 때) 객체를 지우고 다시 삽입하는 대신 객체가 속해 있는 노드의 MBR에 대해서만 갱신을 수행하여 트리 구조의 변경을 지연시킨다. 그리고 객체가 이동한 위치가 MBR내에 위치하는지 확인하는 과정을 빠르게 하기 위하여 해시 테이블로 만들어진 secondary 인덱스 구조체를 이용하여 트리를 이용하지 않고 객체에 접근한다.

<그림 8>은 LUR-Tree 색인을 적용한 위치 조회 알고리즘의 데이터 분포 별 갱신 시간을 나타낸다. 실험 결과 분포 별 갱신 시간은 Static 알고리즘의 경우 Uniform 분포가 48.684초, Gaussian 분포가 30.137초, Skewed 분포는 61.273초로 결과가 나왔다. Static 알고리즘의 경우 고정간격으로 위치조회를 하기 때문에 3개 분포 별로 위치 조회 총수가 167,000 건으로 같지만 갱신 시간은 Skewed 분포가 가장 높고, 그 다음이 Uniform 분포, 그리고 Gaussian 분포가 가장 적게 나왔다. Skewed 분포가 갱신 시간이 가장 높은 이유는 이동체의 분포가 고르지 않고 편향되어 있기 때문에 이동체 위치 갱신 시에 색인의 변경이 좀더 발생한 것이다.

전체적으로 갱신 시간이 적게 나온 이유는 LUR 특성상 객체가 MBR 을 벗어난 경우에도 그 노드의 MBR 을 확장하여 트리구조 변경을 지연시켰기 때문이다. 실험에서는 MBR의 최대 크기를 5 $Km^2$으로 설정하였다.

MATT의 경우 Uniform 분포가 44.946초, Gaussian 분포가 30.489초, Skewed 분포는 50.635초로 결과가 나왔다. MDR 알고리즘의 경우는 Uniform 분포가 48.539초, Gaussian

분포가 28.008초, Skewed 분포는 61.122초로 결과가 나왔다.

이 실험을 통하여 MDR 알고리즘이 Static 알고리즘과 보다 갱신 시간 성능이 우수함을 알 수 있다. 그러나 MDR 알고리즘이 MATT 알고리즘 보다 Gaussian 분포에서는 우수하지만, Uniform 분포 및 Skewed 분포에서는 MATT 갱신 시간 성능이 우수함을 알 수 있다.

<그림 9>는 LUR-Tree 색인을 적용한 위치 조회 알고리즘의 데이터 분포 별 검색 시간을 나타낸다. 이동체 위치 검색 시간은 이동체의 위치 갱신이 발생한 경우, 이동체와 관련된 경보영역들에 이동체가 진입했는지를 판단하기 위하여 경보영역을 질의한 검색 시간이다.

실험 결과 분포 별 검색 시간은 Static 알고리즘의 경우 Uniform 분포가 642.248초, Gaussian 분포가 1036.380초, Skewed 분포는 154.339초로 결과가 나왔다. Static 알고리즘의 경우 고정간격으로 위치조회를 하기 때문에 3개 분포 별로 위치 조회 총수가167,000 건으로 같지만 검색 시간은 Gaussian 분포가 가장 높고, 그 다음이 Uniform 분포, 그리고 Skewed 분포가 가장 적게 나왔다. Skewed 분포가 검색 시간이 가장 적은 이유는 LUR-Tree 특성상 이동체 위치 갱신 시에 트리구조 변경을 최대한 지연 시켰기 때문이다. Skewed 분포의 경우 갱신 시간은 제일 많은 반면 검색시간은 제일 적다. Gaussian 분포의 경우에는 갱신 시간은 적으나 검색 시간은 많다. 즉, LUR-Tree에서는 갱신 시간과 검색 시간과의 Trade-off가 존재 한다.

MATT의 경우 Uniform 분포가 394.703초, Gaussian 분포가 530.108초, Skewed 분포는 98.805초로 결과가 나왔다. MDR 알고리즘의 경우는 Uniform 분포가 583.593초, Gaussian 분포가 513.612초, Skewed 분포는 236.506초로 결과가 나왔다. 이 실험을 통하여 MDR 알고리즘이 Static 알고리즘과 보다 검색 시간 성능이 우수함을 알 수 있다. 그러나 MDR 알고리즘이 MATT 알고리즘 보다 Gaussian 분포에서는 우수하지만, Uniform 분포 및 Skewed 분포에서는 MATT 갱신 시간 성능이 우수함을 알 수 있다.

MDR 알고리즘이 Uniform 분포와 Skewed 분포에서 MATT알고리즘 보다 검색 시간이 약간 많은 이유는 MDR 알고리즘의 버퍼영역과 관련이 있다.

## 7. 결론 및 향후 연구

본 논문에서는 위치기반 경보서비스를 제공하기 위해 대용량의 위치정보를 조회함에 있어서, 이동체의 이동 패턴을 이용하여 불필요한 위치정보 조회 회수를 줄임으로써 시스템의 부하를 줄이는 효율적인 이동체 조회 방법을 제안하였다. 또한 제안한 MDR 알고리즘을 성능을 실험하기 위하여 GSTD 데이터 생성기를 이용하여 Uniform 데이터 분포, Skewed 분포, Gaussian 분포에 따른 정적(Static) 알고리즘, MATT 알고리즘, MDR 알고리즘을 각각 구현하여 비교 실험을 하였다. 또한 이동체 색인구조하에서 성능을 비교하기 위하여 R*-Tree와 LUR-Tree 색인 구조에서 GSTD 데이터 생성기를 이용하여 Uniform 데이터 분포, Skewed 분포, Gaussian 분포에 따른 Static 알고리즘, MATT 알고리즘, MDR 알고리즘을 각각 구현하여 비교 실험을 하였다.

실험 결과 위치기반 경보서비스를 위한 MDR 위치조회 알고리즘은 기존의 정적 알고리즘 과 MATT 보다 위치조회 횟수를 획기적으로 줄였으며, 시스템 성능 면에서 갱신 시간 및 검색 시간이 전반적으로 성능이 훨씬 효율적이다.

앞으로의 연구는 경보서비스를 위한 이동체 데이터를 효과적으로 관리하기 위한 LBA용 이동체 색인구조의 연구가 필요하며, 좀더 시스템의 성능을 향상 시킬 수 있도록 이동통신망에서 제공되는 망 정보 데이터를 이용한 성능 향상 방법이 필요하다.

## 참고문헌


1. Green, J., Betti, D. & Davison, J. : Mobile Location Services: Market Strategies, White Paper. ,Ovum, London, 2000
2. OGC. : OpenGIS Location Services (OpenLS): Core Services, OGC, Wayland, 2004
3. 한기준, "위치기반 서비스(LBS)의 표준화와 연구동향" 정보화정책 제 10권 제 4호, 서울, 2003
4. 3GPP : TS 23.271 V 6.7.0 chapter 9.5, 3GPP, 2004
5. 한국소프트웨어진흥원. "위치기반 서비스 LBS", 한국소프트웨어진흥원 소프트웨어 마켓 뉴스, 서울, 2003
6. Ouri Wolfson, Bo Xu, Sam Chamberlain, Liqin Jiang, "Moving Objects Databases: Issues and Solutions", Proc. of the 10th Int. Conf. on Scientific and Statistical Database Management (SSDBM98), Capri, Italy, July 1-3, 1998, pp. 111-122
7. Dieter Pfoser, Christian S. Jensen. "Capturing the Uncertainty of Moving-Object Representations" , SSD, 1999, pp.111-132
8. Ouri Wolfson, S. Chamberlain, S. Dao, L. Jiang, G. Mendez.," Cost and Imprecision in Modeling the Position of Moving Objects", Proceedings of the Fourteenth International Conference on Data Engineering, ICDE14, 1998,
9. 민경욱, 조대수. "위치기반서비스를 위한 이동체 위치조회 기법", 한국정보처리학회 논문지 Vol.1, 2003,
10. 이진열, 박주훈, 안병익," 위치기반 경보서비스 및 LBS 플랫폼 기술 동향" , 한국정보과학회 학회지, VOL. 23, 2005, PP 75-86
11. 이종연, 안병익, 류근호: "SDE 공간모델의 이력지원 확장", 한국정보처리학회 논문지 A VOL. 05, 1998, PP. 2281-2293 ?
12. 웨이브마켓. "무선통신 네트워크에서 이동체들에 경보 기반 서비스들을 제공하는 시스템". 대한민국 공개특허공보, 출원번호 10-2004-7000651, 서울, 2004
13. Y. Theodoridis, J. R. O Silva and M.A Nascimento, "On the Generation of Spatiotemporal Datasets" , Proc. Of Symposium on Advances in Spatial Databases, 1999, pp.147-164



**안병익**
1993년 동국대학교 대학원 컴퓨터공학과 (공학석사)
2001년 연세대학교 대학원 컴퓨터과학과(박사수료)
1993년~1999년 KT 연구개발본부
2000년~현재 포인트아이(주) 대표이사
관심분야: LBS, 이동체 DB, 공간 DB, 텔레매틱스

**양성봉**
1981년 연세대학교 공학사
1984년 Univ. of Oklahoma 컴퓨터과학과(석사)
1992년 Univ. of Oklahoma 컴퓨터과학과(박사)
1993년~1994년 전주대학교 전자계산학과 전임강사
1994년~현재 연세대학교 컴퓨터과학과 교수
관심분야: P2P 컴퓨팅, 3D 그래픽스