

# 옥내위치기반서비스를 위한 이동객체 데이터베이스 설계

임재걸\*, 정승환\*\*, 권기용\*  
\*동국대학교 컴퓨터멀티미디어학부  
\*\*동국대학교 전자계산학과  
e-mail:{yim\*, honourj\*\*, powergky\*}@dongguk.ac.kr

## Design of a Moving Object Database for Indoor LBS

Jaegel Yim\*, Seunghwan Jeong\*\*, Kiyong Kwon\*

\*Dept of Computer and Multimedia, Dongguk University

\*\*Dept of Computer Science, Dongguk University

### 요 약

위치기반서비스 시스템 구현에 필요한 핵심 요소 기술은 이동객체의 위치를 파악하는 측위 기술인데, 야외에서는 GPS가 실용적인 수준의 정확한 사용자의 위치를 제공하는 수단으로 널리 사용되고 있다. 한편 옥내의 위치기반서비스를 위한 옥내측위방법에 대한 연구도 다양하게 진행되고 있다. 그런데 여러 이동객체의 측위 데이터와 지도 데이터를 가공하여 유용한 정보를 제공하는 위치기반서비스 시스템을 구현하려면 이동객체의 위치와 위치 측정 시각으로 구성된 이동객체 데이터베이스 관리 기술도 역시 없어서는 안 될 핵심기술이며, 이에 대한 연구가 이미 오래 전부터 활발히 수행되었다.

기존의 연구는 대부분 자동차를 이동객체로 생각해왔는데 근래에는 옥내 측위 연구가 활발히 진행되고 있음으로 옥내의 이동객체인 사람을 대상으로 옥내 위치기반서비스가 개발되어야 할 시점이라고 본다. 옥내의 경우에는 지도상의 서비스 영역이 야외의 경우보다 비교할 수 없을 정도로 작고, 이동객체의 이동 속도도 훨씬 느리며, 이동패턴도 규칙성이 떨어지는 등 기존의 연구에서 다른 이동객체의 상황과 매우 상이함으로, 본 논문은 옥내 위치기반서비스를 위한 이동객체 데이터베이스를 설계하고 구현한다.

### 1. 서론

자동차나 보행자와 같이 시간에 따라 위치가 연속적으로 변하거나 전염병이나 태풍의 영향권과 같이 시간에 따라 위치뿐만 아니라 크기나 모양까지도 연속적으로 변하는 시공간객체를 이동객체라 한다[1]. 이동객체의 위치를 고려하여 유용한 정보를 제공하는 것을 위치기반서비스라 한다. GPS(Global Positioning System), 이동전화, 무선네트워크의 발달로 인하여 이동객체의 위치 파악이 용이하게 됨에 따라 위치기반서비스가 활발하게 이루어지고 있다.

근래에 등장하는 상용의 위치기반서비스는 모바일 자원 관리(모바일 노동인력관리, 자동차 위치, 군단관리, 택배, 교통통제, 항공통제)와 위치인지 콘텐츠 전달 서비스로 대별된다. 모바일 자원 관리는 서비스 인력이나 교통 시스템을 추적 및 관리하기 위하여 스케줄된 경로와 함께 위치 데이터를 사용한다. 고객센터와 파송담당자는 모바일 자원 관리 서비스를 사용하여 고객에게 정확한 배달 시간을 알려 주거나, 인력을 최적으로 운용하거나, 긴급 상황에 대처하거나, 기상과 교통체증과 같은 외부 요인에 대처하는 처방을 내린다. 위치인지 콘텐츠 전달 서비스는 모바일 사용자의 위치를 고려하여 정확한 자동차 진행 방향, 가까운 가게의 즉석쿠폰, 주위의 식당, 병원, 무인현금인출기, 주유소와 같은 자원 정보 등 사용자와 관련성이 높은 정보를 제공한다.

상용서비스 외에 군용으로든 위치기반서비스가 이용되는데, 예를 들면, 위치기반서비스는 “xx 지역에 10분 이내에

진입할 예정인 아군 기갑 연대를 검색하라!”와 같은 질의에 답을 제공한다[2].

위치기반서비스 시스템 구현에 필요한 핵심 요소 기술은 이동객체의 위치를 파악하는 측위 기술이다. 야외에서는 GPS가 실용적인 수준의 정확한 사용자의 위치를 제공하는 수단으로 널리 사용되고 있다[3]. 한편 옥내의 위치기반서비스를 위한 옥내측위방법에 대한 연구도 다양하게 진행되고 있다. 그 대표적인 방법에는 적외선 신호를 감지하여 위치를 파악하는 Active Badge[4], 초음파와 RF(Radio Frequency) 신호의 지연시간을 이용하는 Active Bat[5]과 Cricket[6], 신호 세기를 이용한 RADAR[7], 무선근거리통신망용 AP의 신호세기를 이용한 방법[8-12] 등이 있다. 그런데 여러 이동객체의 측위 데이터와 지도 데이터를 가공하여 유용한 정보를 제공하는 위치기반서비스 시스템을 구현하려면 이동객체의 위치와 위치 측정 시각으로 구성된 이동객체 데이터베이스 관리 기술도 역시 없어서는 안 될 핵심기술이며, 이에 대한 연구가 이미 오래 전부터 활발히 수행되었다[1, 2].

그러나 기존의 연구는 대부분 자동차를 이동객체로 생각한다. 근래에는 옥내 측위 연구가 활발히 진행되고 있음으로 옥내의 이동객체인 사람을 대상으로 옥내 위치기반서비스가 개발되어야 할 시점이라고 본다. 옥내의 경우에는 지도상의 서비스 영역이 야외의 경우보다 비교할 수 없을 정도로 작고, 이동객체의 이동 속도도 훨씬 느리며, 이동패턴도 규칙성이 떨어지는 등 기존의 연구에서 다른 이동객체의

상황과 매우 상이함으로, 본 논문은 옥내 위치기반서비스를 위한 이동객체 데이터베이스를 설계한다.

## 2. 관련연구

자동차 관련 서비스를 위한 이동객체 데이터베이스에 관한 연구 결과는 무수히 발표되었다. 이러한 연구를 주제별로 살펴보면 다음과 같다. 첫째, 위치 모델링 방법이다. 위치 모델링 방법에는 점 기반 위치관리와 궤적기반 위치관리가 있는데 현재위치와 현재시각, (l:location, t:time), 순서쌍을 저장하는 점 기반 위치관리 방법은 다음과 같은 단점들이 있다.

- 1) 원하는 시각의 위치가 데이터베이스에 저장되어 있지 않으면 정보를 제공할 수 없다. 예를 들어, 3시 정각에 데이터베이스를 갱신하지 않았다면, "3시 정각에 발생한 사건 현장에서 1 킬로미터 이내에 있던 경관을 찾으라"는 질문에 답을 구할 수가 없다. 더구나 오후 5시 정각에 고객과 가장 가까이 있을 객장 종업원은 누구일까?와 같은 미래 시점 관련 질문에 대한 답을 구하기는 더욱 어렵다.
- 2) 점 기반 위치 관리 방법의 두 번째 단점으로 정확도를 높이고 하면 위치 정보 갱신을 자주해야 하는데, 이것은 엄청난 자원의 낭비를 수반한다.
- 3) 소프트웨어 개발이 불편하다.

궤적기반 위치 관리 방법은 우선 이동객체의 출발점과 목적지를 예측한다. 다음에는 거리와 주행시간이 기록된 전자지도를 이용하여 출발지에서 목적지에 이르는 여행 궤적을 구축한다. 서버는 이동객체로부터 받은 갱신된 위치 정보와 교통 정보 웹 사이트에서 받은 실시간 교통 상황으로 궤적을 유지한다. 이동객체는 GPS에서 수신한 위치 값이 궤적으로 계산한 위치와 한계부정확도 이상 틀릴 경우에만 수신한 위치값을 서버에 송신한다[2].

두 번째 연구 주제는 데이터 접근 연산이다. 연산은 하나의 궤적을 분석하는 연산, 주어진 공간 영역이나 기물과의 관계를 만족하는 궤적을 찾는 연산, 궤적들 간의 어떤 관계를 만족하는 궤적을 찾는 연산 등으로 구분된다.

세 번째 연구 주제는 부정확성 관리다. 이동객체는 항상 움직이기 때문에, 데이터베이스에 기록된 이동객체의 위치는 실제 위치와 일치할 수가 없다. 부정확성 관리는 이러한 부정확성을 설명하고 부정확한 질의를 표현하는데 도움을 주는 것이 목적이다. 부정확도를 낮추려면 이동객체가 위치를 자주 송신해 주어야 하는데 그러면 통신비용과 자원 소비가 증가하게 된다. [13]은 이들 간의 관계를 나타내는 수식을 소개하고, 이를 이용하여 편차비용, 부정확 비용, 통신비용으로 구성된 여행의 비용을 정의함으로써 가장 효율적인 한계 부정확도를 계산하는 방법을 제안한다.

네 번째 연구 주제는 위치 예상이다. 정확한 위치 예상은 적당한 정보를 적기에 제공하기 위하여 중요하다. 인간 행동의 주기성으로부터 집에서 사무실로 출근하였다가 다시 집으로 퇴근하는 것과 같은 행동 패턴을 유추할 수 있고, 행동 패턴을 적용하여 이동객체의 미래의 위치를 예상할 수 있다. 위치 예상에는 교통량이 중요한 요인이 된다. [14]는 교통상황을 예상하는 방법을 소개한다.

다섯 번째 연구 주제는 인덱싱인데, 이동객체 데이터베이스 분야에서 행해지는 연구의 대부분은 인덱싱에 관한 것이다. 특히 궤적에 관한 질의와 갱신을 효율적으로 하기 위하여 수많은 3차원 궤적을 어떻게 인덱싱하는가에 대한

답을 구하기 위한 연구가 대부분이다. 이러한 연구는 [15]처럼 궤적의 각 부분을 라인으로 표현하는 방법이나 [16]처럼 점으로 표현하는 방법으로 대별된다. 대부분의 연구는 가상의 데이터를 다루었는데, 실제 상황에서 이들이 제안한 방법을 검증하는 연구가 필요하다

인간이 위치기반서비스에 노출되면 당사 대두되는 문제가 사생활보호이다. 사생활보호 측면에서는 중앙에서 이동객체를 관리하는 것보다 각 클라이언트가 자신의 위치를 탐지하고 서비스를 제공하는 것이 바람직하다고 [6]은 주장한다. 본 논문은 현재 은행 업무와 전화 통신 서비스처럼 제3의 신뢰하는 기관을 이용하는 방법을 주장한다. [17]은 위치 정보를 다양한 기기로부터 받아 융합하는 문제를 다룬다. 이와 같은 융합은 정확도를 제고하고 어디에서나 위치 정보를 획득할 수 있게 하는 이점이 있다.

본 연구는 옥내위치기반서비스를 위한 이동객체 데이터베이스를 설계한다. 위에서 언급한 모든 기존의 연구 주제들에 대한 답을 제공하는 것이 옥내 이동객체 데이터베이스 연구의 궁극적인 목적이지만 본 논문에서는 우선 간단한 옥내 이동객체 데이터베이스를 구현한다.

## 3. 옥내 이동객체 데이터베이스 설계

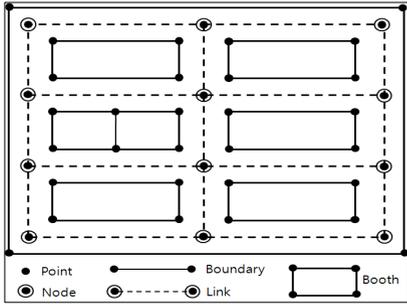
본 장에서는 옥내위치기반서비스를 위한 옥내 이동객체 데이터베이스를 설계한다. 데이터베이스 설계는 일반적으로 사용응도에 의존적으로, 응도에 따라 옥내의 구조, 보행자 통로, 집기, 편의시설, 등 고려해야 할 사항이 무수히 많을 수 있다. 본 논문은 박람회 관람객에게 위치기반서비스를 제공하는 용도라고 가정하며, 관계 데이터베이스 관리 시스템 상에 구현한다고 가정한다. 따라서 박람회장의 구조, 보행자 통로 그리고 관람객의 궤적을 나타내는 테이블들을 설계하고, 이들 간의 공간적 그리고 시간적 관련성을 계산하는 연산자들을 객체지향 프로그래밍 방법으로 설계한다.

본 논문이 설계하는 옥내 이동객체 데이터베이스는 기본적으로 PointTable, NodeTable, LinkTable, BoothTable, BoundaryTable, MovingObjectTable(MOTable이라 칭함), TraceTable 등으로 구성된다. 포인트는 공간적 구조를 모델링하기 위한 기본 요소로써 PointTable에는 노드, 링크, 부스 등에 사용되는 좌표들을 정의할 수 있도록 포인트 ID와 좌표(x와 y)로 구성된다. 노드는 통로의 교차 지점을 지칭하며 NodeTable은 기본적으로 노드의 ID와 포인트 ID, 기하유형 속성으로 구성된다. 링크는 두 노드를 잇는 통로를 지칭하며 LinkTable은 기본적으로 링크 ID, 양끝 포인트 ID, 기하유형 등으로 구성된다. BoothTable은 기본적으로 부스 ID, 부스 이름, 경계 ID(Boundary\_ID)로 구성되고, BoundaryTable은 경계 ID, 경계를 이루는 폴리라인, 기하유형으로 구성된다. MOTable은 이동객체의 ID, 이름 등으로 구성되며, TraceTable은 이동객체의 ID, 시각, 위치를 나타내는 좌표 등으로 구성된다.

본 데이터베이스 구성 요소의 공간적 유형을 살펴보면 점(point), 직선(line), 폴리라인, 다각형 등이므로 공간적 관련성을 연산하기 위하여 Point, Line, Polyline, Arc, Circle, Polygon 등의 클래스를 생성한다. 그리고 다른 객체와 contains, intersects 등의 관련성을 계산하는 메소드를 구현한다.

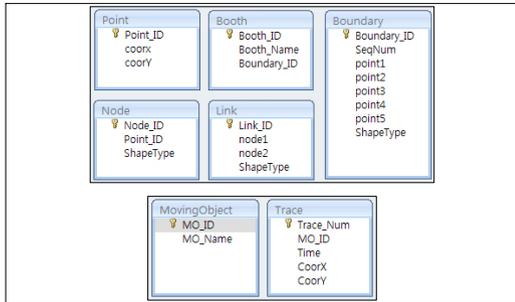
그림 1에 보이는 박람회장의 부스들은 BoothTable에 정의되는데, BoothTable은 BoundaryTable과 PointTable을 통해 부스의 경계를 구분한다. 박람회장의 통로역시

NodeTable, PointTable을 이용하여 LinkTable에 정의된다. 그리고 MovingObjectTable과 TraceTable을 이용하여 이동객체의 실제 움직이는 궤적과 시점을 정의할 수 있다.



(그림 1) 옥내 박람회장 구조의 예

예를 들어 그림 1과 같은 박람회장 구조일 경우에 생성하는 테이블의 구조는 그림 2와 같다.



(그림 2) 옥내 이동객체 데이터베이스의 테이블 구조

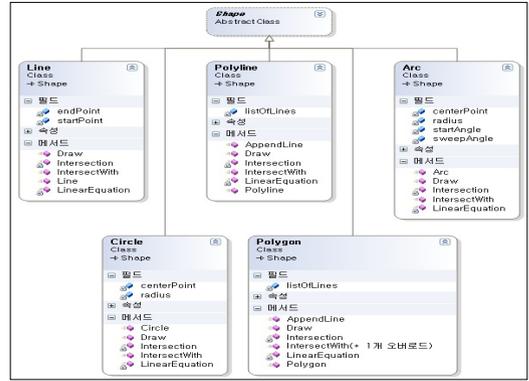
#### 4. 이동객체 컴포넌트 구현

본 장에서는 3장에 정의된 데이터베이스를 이용할 수 있는 이동객체 컴포넌트를 구현한다.

이동객체 컴포넌트는 [11]에 소개된 공간데이터 모델을 기반으로 3장에서 살펴본 이동객체 데이터베이스를 연동하여 구성된다. 여기에 사용된 이동객체의 데이터형과 연산자를 제공하는 인터페이스는 그림 3과 같다.

이 이동객체 컴포넌트는 새로 정의한 DPoint라는 Type을 사용하여 지도정보의 실수 좌표가 반영된 데이터형을 표현하고 있다. 이 DPoint는 기존의 C# 프로그래밍 언어에서 Point Type이 제공할 수 있는 정수(integer)와 실수(float)의 범위를 확장함으로써 double형의 실수를 표현할 수 있다.

그림 3의 이동객체 컴포넌트는 공간 데이터에 대하여 점 객체(DPoint), 단순 선 객체(Line, Arc), 복합 선 객체(Polyline), 면 객체(Polygon, Circle)에 대한 각 데이터형을 처리할 수 있다. 예를 들어 그림 1과 같은 옥내 구조에 대하여 특정 좌표나 Node는 DPoint 객체로 처리하고 Link는 Line 객체로 처리하며, Booth는 Polygon 객체를 이용하여 표현할 수 있다. 그리고 움직이는 이동객체의 실시간 궤적은 Polyline 객체로 표현한다. 이동객체 컴포넌트가 표현할 수 있는 Line, Arc, Polyline, Polygon, Circle 객체들 간의 교차 관계는 IntersectWith() 메소드를 통해 답을 구할 수 있는데, 특히 선 객체가 면 객체에 대해서 진입, 내재, 통과 여부를 판별할 수 있다. 여기서 선



(그림 3) 이동객체 컴포넌트의 다이어그램

<표 1> 파라메타로 오는 두 Line의 교차여부 판별 알고리즘

```

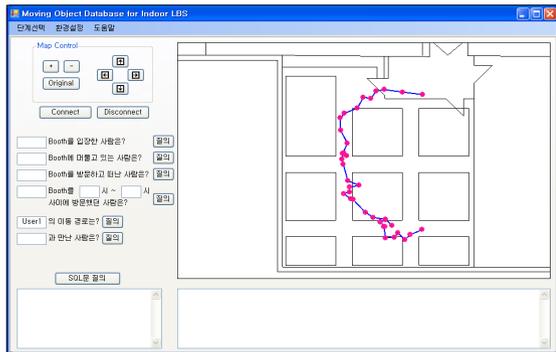
Algorithm Intersection(sPos1, ePos1, sPos2, ePos2,
    _coorX, _coorY)
sPos1과 ePos1은 첫 번째 Line에 대한 시작, 끝 좌표
sPos2와 ePos2는 두 번째 Line에 대한 시작, 끝 좌표
_coorX와 _coorY는 Line1과 Line2가 교차할 경우 이
교점의 좌표가 저장된다.
Line1과 Line2가 교차하면 True, 아니면 False를 반환한다.
1. existIntersection = false;
   // 두 직선의 교점이 없거나 일치하는 경우.
2. if (sPos1.X == ePos1.X && sPos2.X == ePos2.X)
   2.1 existIntersection = false;
   2.2 return existIntersection;
3. else if (sPos1.X == ePos1.X && sPos2.X != ePos2.X)
   3.1 existIntersection = true;
   3.2 _coorX = sPos1.X;
   // Line2의 기울기(gradient)와 Y 절편(yIntercept)을 구함.
   3.3 LinearEquation(sPos2, ePos2, gradient, yIntercept);
   3.4 _coorY = gradient * _coorX + yIntercept;
4. else if (sPos1.X != ePos1.X && sPos2.X == ePos2.X)
   4.1 existIntersection = true;
   4.2 _coorX = sPos2.X;
   // Line1의 기울기(gradient)와 Y 절편(yIntercept)을 구함.
   4.3 LinearEquation(sPos1, ePos1, gradient, yIntercept);
   4.4 _coorY = gradient * _coorY + yIntercept;
5. else
   5.1 LinearEquation(sPos1, ePos1, gradient1, yIntercept1);
   5.2 LinearEquation(sPos2, ePos2, gradient2, yIntercept2);
   5.3 if (gradient1 != gradient2)
       5.3.1 existIntersection = true;
       5.3.2 _coorX = (yIntercept2 - yIntercept1)
           / (gradient1 - gradient2);
       5.3.3 _coorY = _coorX * gradient1 + yIntercept1;
   5.4 else
       5.4.1 existIntersection = false;
       5.4.2 return existIntersection;
   5.5 end if
6. end if
7. if (_coorX가 Line1과 Line2의 X좌표 범위 안에 있다 ||
    _coorY가 Line1과 Line2의 Y좌표 범위 안에 있다)
   7.1 existIntersection = true;
8. else
   8.1 existIntersection = false;
9. end if
10. return existIntersection;
End Algorithm
    
```

객체가 면 객체를 한번이라도 교차한다면 진입상태인 것이고, 그 교차 횟수가 홀수번일 경우는 내제한 상태, 짝수

번일 경우는 통과한 상태이다. 이 IntersectWith() 메소드는 호출되는 파라미터에 따라 질의의 결과가 달라지며, 이 메소드는 각 객체들을 Line 객체 단위로 분할한 후 Intersection() 메소드를 통해 각 Line간의 교차여부를 판별하게 된다. 표 1은 Intersection() 메소드가 파라미터로 오는 두 Line의 교차여부를 판별하는 알고리즘이다. 이 알고리즘은 두 Line의 시작점과 끝점들을 파라미터로 받는데, 만약 두 Line이 교차할 경우 이 교점을 (\_coorX, \_coorY)에 저장한 후 True를 반환하며 그렇지 않으면 False를 반환한다. 이 알고리즘의 step 1 ~ step 6은 파라미터로 받은 두 Line을 두 개의 직선방정식으로 변환하여 교점을 구하는 과정이고, step 7 ~ step 9은 앞서 구한 교점이 실제 두 Line의 X좌표와 Y좌표 사이에 존재하는지(실제로 교차하는지)를 판별하는 과정이다.

**5. 실험**

실험을 위해 3장에서 설계한 이동객체 데이터베이스는 Oracle 10g가 설치된 SPAR 서버에 구현하였고, Lap-Top 컴퓨터에서 [8-12]에 소개한 옥내 추적 시스템과 이동객체 컴포넌트를 구현하였다. 본 논문에서는 간단한 옥내 이동객체 데이터베이스를 구현하기 위해 옥내 공간데이터들을 이동객체 데이터베이스에 갱신하였고, 사용자가 움직이는 실시간 이동 궤적들은 다양한 시간과 많은 이동객체들이 반영된 가상의 데이터들을 사용하였다.



(그림 4) 이동객체 컴포넌트와 이동객체 데이터베이스가 연동된 실행화면

그림 4는 이동객체 컴포넌트와 이동객체 데이터베이스가 연동된 실행화면으로, 어플리케이션 좌측에 대표적인 질의들 몇 가지를 바로 수행 할 수 있도록 구성하였다. 그림 4는 User1의 이동 경로를 질의한 결과로 이동객체 데이터베이스에서 User1에 해당하는 trace정보들을 이동객체 컴포넌트를 이용하여 지도파일에 출력한 결과로써, 이 외에 다음과 같은 간단한 질의들이 올 수 있다.

- 한국 Booth를 입장한 사람은?
- 한국 Booth에 머물고 있는 사람은?
- 한국 Booth를 방문하고 떠난 사람은?
- 한국 Booth를 1시 ~ 3시 사이에 방문했던 사람은?
- User3과 만난 사람은?

또한 실험을 위해 상황에 따른 질의가 가능하도록 SQL문 질의를 사용할 수 있도록 구성하였다.

**6. 결론**

본 논문은 대부분 자동차가 이동객체의 대상이었던 기존 연구에 반하여 근래에 옥내 측위 연구가 활발히 진행되고 있는 옥내의 이동객체인 사람을 대상으로 옥내 이동객체 데이터베이스 설계에 대하여 살펴보았다. 실험을 위해 이동객체 컴포넌트를 구현 하였고, 간단한 질의 실험을 통해 설계한 옥내 이동객체 데이터베이스의 구현을 보여주었다. 하지만 가상의 실시간 이동 궤적을 사용한 제한된 구현이었다는 단점이 있다. 향후에는 옥내 측위 시스템과 이동객체 컴포넌트, 그리고 이동객체 데이터베이스의 실시간 연동이 가능하도록 많은 연구가 필요하다.

**참고문헌**

- [1] 김경숙, 정성훈, 이기준, “이동객체를 위한 데이터베이스” 한국인터넷 정보학회 제2권 제4호, pp. 21-36
- [2] Orii Wolfson, “Moving Objects Information Management: The Database Challenge,” The Proceedings of the 5th Workshop on Next Generation Information Technologies and Systems (NGITS'2002), Dan Caesarea Hotel, Caesarea, Israel, June 25-26, 2002.
- [3] 주재훈, 염경환, 이상경, “GPS/ GLONASS 통합 수신용 RF 전단부의 설계 및 제작,” 정보과학회논문지, 제12 권 4호, 2001, pp. 1226-2285.
- [4] Want, R., A. Hopper, V. Falcao, and J. Gibbons, “The Active Badge Location System,” ACM Transactionson Information Systems, Vol. 10, No.1(1992) .pp. 91-102.
- [5] Harter, A. and A. Hopper, “A New Location Technique for the Active Office,” IEEE Personal Communications, Vol.4, No.5(1997), pp.43-47.
- [6] Priyanthat, N., A. Chakraborty, and H. Balakrishnan, “The Cricket Location-Support System,” Proc. of 6th ACMInternationalConferenceonMobileComputingandNetworking, Boston, MA, Aug. 2000.
- [7] Bahl, P. and V. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System,” INFOCOM2000, Mar.2000, pp.775-784
- [8] Yim, J. “Introducing a decision tree-based indoor positioning technique,” Expert Systems with Applications, Vol. 34, Issue 2, 2008, pp. 1296-1302.
- [9] 임재걸 “옥내 측위를 위한 지문 방식 알고리즘들의 성능 분석,” 전자공학회논문지 제 43권 CI 편 제6호, 2006년 11월 pp. 454-462.
- [10] 임재걸, 심규박, 정승환, “RSSI 판독 라이브러리 함수 및 옥내 측위 모듈 구현,” 멀티미디어학회 논문지 제 10권 11호, 2007년 11월 pp. 1483-1495.
- [11] 임재걸, 주재훈, 정승환, “위치기반서비스를 위한 지도정보가 반영된 옥내측위 통합 시스템,” 한국정보시스템학회, 17권 1호, (2008년 3월 30일) pp. 131-154
- [12] 임재걸, 박찬식, 주재훈, 정승환, “Wi-Fi 기반 옥내측위를 위한 확장칼만필터 방법,” Journal of Information Technology Applications and Management Vol. 15 No. 2 June 2008, pp. 51-65.
- [13] O. Wolfson, A. P. Sistla, S. Chamberlain, Yelena Yesha. “Updating and Querying Databases that Track Mobile Units,” Distributed and Parallel Databases, 7, 257-287, 1999
- [14] S. Handley, P. Langley, F. Rauscher, “Learning to Predict the Duration of an Automobile Trip,” Proc. of the 4th International Conference on Knowledge Discovery and Data Mining (1998).
- [15] S. Saltenis, C.S. Jensen, S.T. Leutenegger, M.A. Lopez: “Indexing the Positions of Continuously Moving Objects,” ACM SIGMOD Conference 2000.
- [16] A.K. Agarwal, L. Arge, J. Ericskon: “Indexing moving points”. Proc. Of ACMPODS 2000 conference.
- [17] J. Myllymaki, S. Edlund, “Location Aggregation from Multiple Sources,” Proc. Of the 3rd Int. Conference on Mobile Data Management (MDM), Singapore, Jan. 02.