

# 도로 네트워크 환경에서 이동 객체 위치 예측을 위한 효율적인 인덱싱 기법<sup>†</sup>

## An Efficient Indexing Technique for Location Prediction of Moving Objects in the Road Network Environment

홍동숙\* / Dong-Suk Hong 김동오\*\* / Dong-Oh Kim

이강준\*\*\* / Kang-Jun Lee 한기준\*\*\*\* / Ki-Joon Han

### 요약

현재 무선 통신 기술과 위치 정보 기술의 발달은 다양한 위치 기반 서비스(LBS: Location Based Services)의 발전을 가져왔으며, 위치 기반 서비스에서 이동 객체의 미래 위치를 빠르게 예측하기 위한 미래 인덱스의 필요성이 높아지고 있다. 미래 인덱스와 관련한 대표적인 연구로써 도로 네트워크 환경에서 이동 객체의 과거 궤적 정보를 이용하여 신뢰성을 높인 확률 궤적 예측 기법이 연구되었다. 그러나, 이 기법은 장기간 미래 질의 시 방대한 미래 궤적 탐색 부하로 인해 예측 성능이 떨어지게 되며, 이 때문에 발생하는 빈번한 미래 궤적 갱신으로 인해 인덱스 유지비용이 매우 높아지게 된다.

따라서, 본 논문에서는 효율적인 장기간 미래 위치 예측을 위한 셀 기반의 미래 인덱싱 기법인 PCT-Tree(Probability Cell Trajectory-Tree)를 제시한다. PCT-Tree는 방대한 과거 궤적의 확률을 셀 단위로 재구성함으로써 인덱스 크기를 줄이고, 장기간 미래 질의의 예측 성능을 개선시킨다. 또한, 과거 궤적 정보를 이용하여 신뢰성있는 미래 궤적을 예측함으로써 미래 궤적 예측 오류에 따르는 통신 비용과 미래 궤적 갱신으로 인한 인덱스 재구성 비용을 최소화 할 수 있다. 실험을 통해 도로 네트워크 환경에서 PCT-Tree가 기존 인덱싱 기법보다 장기간 미래 질의 성능이 우수함을 입증하였다.

### Abstract

The necessity of future index is increasing to predict the future location of moving objects promptly for various location-based services. A representative research topic related to future index is the probability trajectory prediction technique that improves reliability using the past trajectory information of moving objects in the road network environment. However, the prediction performance of this technique is lowered by the heavy load of extensive future trajectory search in long-range future queries, and its index maintenance cost is high due to the frequent update of future trajectory.

† 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음.

■ 논문접수 : 2006.12.20      ■ 심사완료 : 2007.3.26

\* 교신저자 건국대학교 대학원 컴퓨터공학과 박사과정 (dshong@db.konkuk.ac.kr)

\*\* 건국대학교 대학원 컴퓨터공학과 박사과정 (dokim@db.konkuk.ac.kr)

\*\*\* 건국대학교 대학원 컴퓨터공학과 박사과정 (kjlee@db.konkuk.ac.kr)

\*\*\*\* 건국대학교 컴퓨터공학과 교수(kjhan@db.konkuk.ac.kr)

Thus, this paper proposes the Probability Cell Trajectory-Tree (PCT-Tree), a cell-based future indexing technique for efficient long-range future location prediction. The PCT-Tree reduces the size of index by rebuilding the probability of extensive past trajectories in the unit of cell, and improves the prediction performance of long-range future queries. In addition, it predicts reliable future trajectories using information on past trajectories and, by doing so, minimizes the cost of communication resulting from errors in future trajectory prediction and the cost of index rebuilding for updating future trajectories. Through experiment, we proved the superiority of the PCT-Tree over existing indexing techniques in the performance of long-range future queries.

**주요어 :** 이동 객체, 미래 인덱스, 미래 궤적, 셀, 확률 매트릭스

**Keyword :** Moving Objects, Future Index, Future Trajectory, Cell, Probability Matrix

## 1. 서론

최근 무선 통신 기술과 위치 정보 기술의 발달은 다양한 위치 기반 서비스 (LBS: Location Based Services)의 발전을 가져왔다. 또한, 개인의 휴대용 전자 장치 성능이 급속도로 발전되고 있으며, 이와 연계된 위치 기반 서비스의 요구가 증대되고 있다 [1,2,3]. 이러한 위치 기반 서비스에서 클라이언트 인 이동 객체는 자신의 위치 정보를 서버에 전송하고, 서버는 이들 각각의 위치 정보를 수집 및 인덱싱하여 필요한 위치 정보를 검색해 준다. 이와 같은 위치 기반 서비스를 위한 이동 객체의 위치 인덱스의 필요성이 나날이 증가되고 있으며, 특히 물류 운송, 차량 추적, 응급 상황 시 차량의 현재 및 미래 위치를 빠르게 서비스하기 위한 미래 인덱스에 대한 연구가 활발하게 진행되고 있다 [4,5,6,7].

미래 인덱스는 이동 객체의 현재 및 미래 위치를 질의하기 위한 시공간 인덱스이다. 즉, 미래 인덱스는 현재 시점의 이동 객체 위치에 대한 효율적인 검색뿐만 아니라 미래 시점의 이동 객체 위치에 대한 효율적인 예측을 위한 것이다. 예를 들어, “앞으로 10분 뒤 시청 앞을 지나 갈 택시는?”이나 “A 이동 객체의 10분 뒤 위치는?”과 같은 이동 객체의 미래 위치를 얻기 위한 질의 처리 시 미래 인덱스가 사용된다 [6]. 미래 인덱스는 인덱싱할 미래 질의 시간의 범위에 따라 인덱스의 크기 및 성능 부

하에 많은 영향을 준다 [8]. 일반적으로 질의 시간이 길수록 인덱스의 크기는 커지며, 질의의 정확성은 떨어지게 된다. 특히, 이런 장시간의 미래 인덱스 질의는 불확실성으로 인해 미래 위치의 오차를 증가시키며, 또한 인덱스 구성의 부하를 야기한다 [9,10].

특히, 미래 인덱스는 도로 네트워크를 사용하는 지 여부에 따라 크게 네트워크 기반의 미래 인덱스와 비네트워크 기반의 미래 인덱스로 나뉘어진다. 도로 네트워크 환경에서는 이동 객체는 항상 도로 세그먼트 위를 움직인다고 전제되며, 이때 이동 객체가 위치한 도로 세그먼트와 속도를 이용할 경우 보다 정확한 이동 객체의 미래 위치 예측이 가능하기 때문에 도로 네트워크 환경에서는 네트워크 기반의 미래 인덱스가 효과적이다 [11]. 네트워크 기반 미래 인덱스에서는 이동 객체의 미래 위치를 예측하기 위해서 먼저 이동 객체의 미래 궤적을 예측하는 것이 필요하다 [1,12].

이러한 네트워크 기반 미래 인덱스로는 이동 객체가 현재 위치한 도로 세그먼트 정보를 이용하여 이동 객체의 위치를 예측하는 세그먼트 트래킹 기법 [5], 적은 세그먼트 변화가 일어나게 도로 세그먼트를 확장하여 도로 네트워크를 변경하는 도로 네트워크 수정 기법 [4], 교차점에서 미래에 지나갈 도로 세그먼트를 확률적으로 예측하여 미래 궤적을 생성하는 PLM(Prediction Location Model) [6]

이 연구되었다. 그러나 세그먼트 트래킹 방식은 예측할 도로 세그먼트 수가 증가할수록 미래 위치 예측의 신뢰성이 현저히 저하되며, 이로 인해 서버에서 현재 위치 정보를 갱신하기 위한 통신 비용 및 미래 인덱스에 대한 갱신 비용이 증가한다. 도로 네트워크 수정 기법은 도로 네트워크의 일반적인 속성과 일반적인 이동 객체의 특성만을 고려하기 때문에 개개의 이동 객체의 기호와는 거리가 멀어 실제 상황에서는 많은 도로 세그먼트의 변화를 가져오는 단점이 있다. PLM은 장기간 예측일 경우 관리할 확률 테이블이 많아지며 확률 검색 트리가 방대해져 미래 위치 예측 시간이 느려지는 문제점이 있다[13].

본 논문에서는 이러한 미래 인덱스의 문제점을 해결하기 위해 도로 네트워크 환경에서 이동 객체 위치 예측을 위한 효율적인 미래 궤적 인덱싱 기법인 PCT-Tree(Probability Cell Trajectory-Tree)를 제시한다. PCT-Tree는 장기간의 미래 질의 시 미래 위치 예측 시간이 느려지는 문제점을 해결하기 위해 셀 기반의 미래 궤적 예측과 미래 위치 검색 방법을 지원하는 미래 인덱스이다. PCT-Tree는 과거 이동 객체의 셀 내부 진입, 진출 정보를 확률 매트릭스로 관리하여, 이를 통해 보다 신뢰성있는 미래 위치를 예측한다. 또한, PCT-Tree는 장기간 미래 예측 시 확률 변수 순회 방식을 이용하여 보다 빠른 미래 궤적 예측 성능을 보장한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 기존의 네트워크 기반 미래 인덱싱 기법에 대하여 분석한다. 제 3 장에서는 이동 객체의 미래 위치 예측을 위한 셀 기반의 미래 인덱스인 PCT-Tree에 대해 설명하고, 제 4 장에서는 실험을 통한 성능 평가 결과를 알아본다. 마지막으로 제 5 장에서는 결론에 대해 언급한다.

## 2. 네트워크 기반 미래 인덱싱

본 장에서는 이동 객체의 미래 위치를 예측할 수 있는 기존의 네트워크 기반 미래 인덱싱 기법들에

대해 분석한다.

### 2.1 세그먼트 트래킹 기법

세그먼트 트래킹의 기본적인 의도는 클라이언트가 이동 시에 도로 네트워크의 정보를 활용하는데 있다. 서버는 클라이언트로부터 받은 정보를 이용하여 클라이언트가 어느 도로 네트워크의 세그먼트에 있는지 알 수 있으며, 이를 위해 서버는 맵 매칭 기술을 사용한다. 세그먼트 트래킹에서 클라이언트의 미래 위치는 폴리라인(Polyline)으로 표현된 세그먼트를 따라 일정 속도로 움직이는 점으로 예측될 수 있다[5]. 이때, 속도는 최근에 보고된 것을 사용한다. 예측된 위치가 세그먼트의 끝에 도달할 경우 세그먼트 트래킹은 점 트래킹 방식으로 전환된다. 만약 도로의 유실이나 이동 객체가 도로 이외의 장소를 움직일 경우 세그먼트 트래킹은 임시로 벡터 트래킹으로 전환하게 되며, 다음 클라이언트의 위치보고 시 서버는 다시 세그먼트 트래킹 방식으로 위치를 예측하게 된다.

그러나 세그먼트 트래킹의 경우 세그먼트의 끝 부분에서는 더 이상의 미래 위치를 예측하지 못해 점이나 벡터 트래킹을 보완해야 하고, 또한 현재 세그먼트에서 다른 세그먼트로 변경 시 마다 서버에 보고가 일어나는 문제점을 가지고 있다.

### 2.2 도로 네트워크 수정 기법

도로 네트워크 수정 기법(Road Network Modification)은 세그먼트 트래킹 기법의 문제점을 보완하기 위해 보다 적은 세그먼트 변화가 일어나게 도로 세그먼트를 확장하여 도로 네트워크를 변경하는 기법이다. 도로 네트워크 수정의 목적은 이동 객체가 도로 네트워크를 이동할 때 가능한 적은 세그먼트 변화를 가지도록 세그먼트를 연결하는 것이다[4]. 이와 같은 도로 네트워크 수정 기법은 세그먼트 트래킹 기법에 비해 효과적인 미래 위치를 예측하는 결과를 얻을 수 있다. 미래 인덱스에서 예측된 세그먼트의 변화가 적어지면 통신 비용

및 서버의 부하를 동시에 줄일 수 있으며, 도로 네트워크 환경에서 도로의 속성 및 이동 객체의 특성을 이용한 도로 네트워크 수정은 운영 시 별도의 예측 알고리즘 부하 없이 도로 네트워크 환경에 적합한 미래 궤적을 예측하는 장점을 가지고 있다.

그러나, 이와 같은 기법은 도로 네트워크의 일반적인 속성(거리 번호, 주 도로, 간선 도로)과 일반적인 이동 객체의 특성(직진성)만을 고려하기 때문에 개개의 이동 객체의 기호와는 거리가 멀어 실제 상황에서는 많은 도로 세그먼트의 변화를 가져오는 단점이 있다.

### 2.3 PLM의 확률 매트릭스 기법

PLM에서는 서버는 과거 궤적 정보를 통해 이동 객체가 교차점에서 선택한 도로 세그먼트의 빈도를 가지고 확률 매트릭스(Probability Matrix)를 관리한다[6]. 이동 객체의 과거 궤적에 대한 확률 매트릭스가 서버에 저장된 후 이동 객체는 미래 질의를 서버에게 요구하게 된다. DCW(Dynamic Computational Window)는 이동 객체가 전체 도로 네트워크 환경에서 질의 영역을 제한한 영역인데, DCW의 반경은 이동 객체의 속도와 예측 시간 간격으로 정의될 수 있다. PLM은 서브 네트워크와 DCW에 의해 결정된 진출점(Exit Points)을 사용하여 이동 객체의 위치에서 예측 가능한 확률 검색 트리를 생성하게 되며, 이를 이용하여 이동 객체의 현재의 위치에서 앞으로 지나갈 도로 세그먼트의 연속인 미래 궤적을 예측한다. PLM은 주기성을 가진 이동 객체의 미래 궤적을 예측하는데 매우 효과적인 확률적 접근 방법을 제시하고 있으며, 이와 같은 방법은 교차점에서 이동 객체의 기호에 따른 미래 도로 세그먼트의 예측을 가능하게 한다.

그러나 PLM은 매 교차점마다 이동 객체별 확률 매트릭스를 관리해야 되기 때문에 확률 변수의 수가 많아지며, 또한 예측 시간이 길어질수록 확률 검색 트리가 방대해지고 미래 궤적 예측을 위한 검색 시간이 길어지는 문제점이 있다.

### 3. PCT-Tree

본 장에서는 이동 객체의 장기간 미래 위치 예측을 위해 미래 궤적을 생성하는 셀 기반 미래 인덱스인 PCT-Tree의 전체 구조, PCT-Tree의 셀 분할, 셀 확률 매트릭스, 그리고 PCT-Tree의 알고리즘에 대해 설명한다.

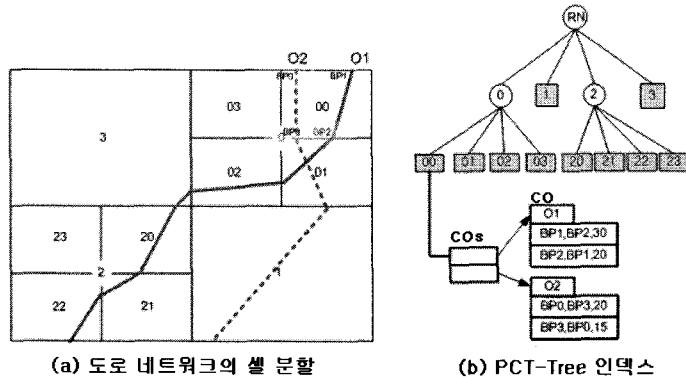
#### 3.1 PCT-Tree의 전체 구조

도로 네트워크가 주어지면 PCT-Tree는 영역 분할을 수행하여 비균등 셀들을 생성한다. PCT-Tree의 분할 조건은 도로 세그먼트 수와 셀에 진입, 진출하는 도로 세그먼트 수를 기준으로 하는데, 이렇게 분할된 셀들은 객체마다 셀 확률 변수를 유지한다. 이동 객체의 과거 궤적이 서버로 전송되면 과거 궤적이 지나는 셀들의 셀 확률 매트릭스의 값이 갱신되며 객체의 과거 이력에 대한 경험치가 셀 확률 변수에 쌓이게 된다.

PCT-Tree에서는 이동 객체 궤적을 셀 단위로 단순화시켜 객체의 확률 검색 트리를 축소함으로써 인덱스 크기를 줄일 수 있고, 또한 미래 위치 질의 시간이 길어질수록 미래 궤적 예측 시간이 느려지는 단점을 해결한다. PCT-Tree에서 이동 객체 궤적은 한 셀에서 다른 셀로의 진입 및 진출한 위치들의 연속으로 표현될 수 있는데, 이를 셀 궤적이라 부른다. 이 경우 셀 영역 내부의 도로 세그먼트의 정보들이 사라지는 문제점을 가지는데, PCT-Tree는 셀 영역 내의 진입, 진출 지점을 알고 있으므로 셀 내부 궤적은 가장 빠른 길을 이용하는 이동 객체의 행동 유형에 근거하여 최단 경로 알고리즘을 통해 생성된다.

PCT-Tree는 Quad-Tree 자료 구조를 가지며, 미래 위치 예측을 위해 PCT-Tree 단말 셀마다 셀 확률 매트릭스(CPM:Cell Probability Matrix)를 구성하는 셀 확률 변수를 가진다. 그림 1은 PCT-Tree의 구조의 예를 보여준다.

그림 1(a)에서와 같이 객체 O1, O2의 과거 궤



<그림 1> PCT-Tree의 구조

적은 여러 개의 셀을 지나게 된다. 이때, 셀 내부의 도로와 셀 영역이 만나는 진입 경계점과 진출 경계점이 생성된다. 이렇게 과거 궤적은 셀과 만나면서 생기는 경계점, 진입시간, 진출 시간으로 구성되는 셀 궤적으로 변환된다. O1의 셀 궤적은 00(1,2,t0,t1) → 01(1,3,t1,t2) → 02(0,1,t2,t3) → 3(0,1,t3,t4) → 20(0,1,t4,t5) → 21(0,1,t5,t6) → 22(0,1,t7,t8)이며, O2의 셀 궤적은 00(0,3,t2,t3) → 01(0,2,t3,t4) → 1(0,1,t4,t5)이다. 이렇게 변환된 셀 궤적은 PCT-Tree에 삽입된다. O1의 셀 궤적을 보면 셀 00에서 BP1로 t0시간에 진입한 후 BP2로 t1시간에 진출하는 것을 알 수 있다. 이러한 정보를 가지고 셀 객체 CO 레코드를 생성한다. CO 레코드는 객체 ID, 진입 경계점, 진출 경계점, 지나간 횟수를 가진다.

그림 1(b)는 주어진 도로 네트워크의 셀 궤적이 입력된 PCT-Tree 인덱스를 보여준다. 그림 1(a)에서 셀 00에 대해 O1이 진입하고 진출할 수 있는 경우의 수는 BP1→BP2, BP2→BP1의 2개이다. 그림 1(b)에서 보는바와 같이 객체 O1은 셀 00에서 BP1→BP2로는 30번 방문하고, BP2→BP1으로는 20번을 방문한 것을 알 수 있다. 마찬가지로 객체 O2는 셀 00에서 BP0→BP3으로는 20번 방문하고, BP3→BP0으로는 15번 방문하였다. 이렇게 객체의 과거 궤적이 입력될수록 셀 객체의 확률 변수는 많은 경험을 저장하게 되며, 미래 궤적

예측 시 셀 확률 매트릭스를 구성하는 변수로 사용된다. 예를 들어, 객체 O1이 셀 00의 BP1에서 특정 미래 질의 시간까지의 미래 궤적을 요청하면, PCT-Tree는 셀 객체 내부의 셀 진입, 진출 정보를 이용하여 셀 확률 변수를 구한 후 질의 시간 범위의 셀 확률 매트릭스를 순회하여 가장 확률 높은 궤적을 예측하여 반환한다.

### 3.2 PCT-Tree의 셀 분할

도로 네트워크 환경에서 이동 객체가 지나갈 미래 경로와 미래 시간을 미래 궤적이라 부른다. PCT-Tree는 주어진 도로 네트워크에 대해서 비균등 크기의 셀 단위 분할하여 초기 공간 인덱스를 구성한다. 그림 1(a)는 주어진 도로 네트워크를 비균등 셀 단위로 분할한 예를 보여준다. 이때, 생기는 단말 셀은 셀 궤적에 사용되는데, 셀의 크기가 커지면 미래 위치 예측 시 위치 오차가 커지게 되고, 너무 작으면 인덱스 크기의 증가 및 검색 속도가 느려지게 된다. 이와 같이 셀 크기는 PCT-Tree에 큰 영향을 미치는데, PCT-Tree의 셀 크기는 PCT-Tree 생성 시 셀 분할 조건에 의해 결정된다.

PCT-Tree는 셀 내부의 도로 세그먼트 수와 셀로 진입/진출하는 경계점의 수를 이용하여 셀을 분할한다. 즉, 두 가지 셀 분할 임계치가 있는데, 하나

는 내부 도로 세그먼트 수이고 다른 하나는 내부 도로 세그먼트와 셀의 경계가 만나서 생기는 경계점 수이다. 셀의 크기가 너무 커지면, 셀 당 관리할 도로 세그먼트 수가 커지며, 이 경우 내부 궤적에 대한 관리 비용이 늘어나게 된다. PCT-Tree의 단말 셀에서 경계점은 도로 세그먼트와 셀의 영역이 만나는 점으로써 해당 셀의 진입, 진출 지점을 의미한다. 이 경우 경계점이 많다는 것은 관리될 진입, 진출 지점이 늘어나게 되는 것을 뜻하며, 진입, 진출 지점의 증가는 PCT-Tree 셀 확률 매트릭스 크기의 증가를 가져온다.

PCT-Tree는 이동 객체마다 셀 내부의 도로 세그먼트를 선택하여 진입하고 진출하는 경험을 셀 확률 매트릭스의 확률 변수에 저장한다. 셀 내부의 경계점 수는 확률 변수의 수를 결정하는데, 경계점 수가 16미만일 경우 최대 확률 변수의 수 (진입 경계점 수 \* 진출 경계점 수)는 256미만으로 1바이트로 표현이 되며, 16이상일 경우는 1바이트 이상으로 표현이 된다. 그러므로 PCT-Tree의 셀 확률 변수의 크기는 경계점 수를 16미만으로 정하는 것이 전체 인덱스 크기의 효율을 높일 수 있다. 따라서 PCT-Tree의 셀 분할 임계치는 경계점 수가 16미만일 경우 실험에 의해 최소 셀 수를 가지는 경계점 수와 도로 세그먼트 수를 이용하여 구할 수 있다.

### 3.3 셀 확률 매트릭스

PCT-Tree는 장기간 미래 궤적 예측을 위해 셀 확률 매트릭스를 이용한다. 셀 확률 매트릭스는 이동 객체의 미래 궤적을 생성하기 위한 셀 확률 변수를 가지고 있다. 셀 확률 변수는 이동 객체가 지나간 경계점만을 가지고 생성되며, 이러한 셀 확률 변수를 이용하여 미래 궤적 예측을 위한 탐색이 수행된다. n개의 진입, 진출 경계점을 가진 셀 c에서 식 1은 셀 확률 매트릭스를 보여 준다.

식 1에서 시간  $t_k$ 는 현재 시간을 뜻하며,  $B_1, B_2, \dots, B_n$ 은 셀에 진입하고 진출하는 경계점을 의미한다.  $P(B_i|B_j)$ 은 이동 객체가 현재 셀의 경계점

$B_1$ 으로 진입한 후  $B_1$ 으로 진출할 확률을 의미하며,  $P(B_2|B_1)$ 은  $B_1$ 으로 진입한 이동 객체가  $B_2$ 로 진출할 확률을 의미한다. 이와 같은 진입, 진출 데이터가 셀 확률 매트릭스의 확률 변수로 사용된다.  $P(B_i|B_j)(i=1, n)$ 는 빈도를 이용하여 식 2와 같이 표현될 수 있다.

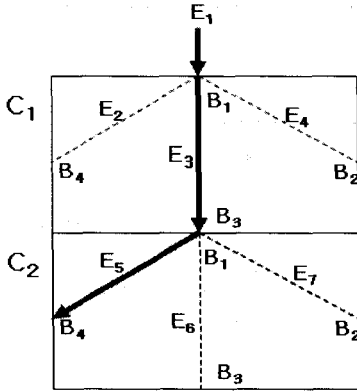
$$CPM(C, t_k) = \begin{pmatrix} P(B_1|B_1) & P(B_2|B_1) & \dots & P(B_n|B_1) \\ P(B_1|B_2) & P(B_2|B_2) & \dots & P(B_n|B_2) \\ \dots & \dots & \dots & \dots \\ P(B_1|B_n) & P(B_2|B_n) & \dots & P(B_n|B_n) \end{pmatrix} \quad (식 1)$$

$$\begin{aligned} P(B_1|B_1) &= P(N_{1,1}/N_1) \\ P(B_2|B_1) &= P(N_{2,1}/N_1) \\ &\dots \\ P(B_n|B_1) &= P(N_{n,1}/N_1) \end{aligned} \quad (식 2)$$

식 2에서  $N_i$ 는  $B_i$ 에 방문한 횟수를 나타내며,  $N_{i,j}$ 는  $B_j$ 에서  $B_i$ 으로 방문한 횟수를 나타낸다. 셀 내부에서 진입한 도로 세그먼트  $B_1$ 에서 나갈 수 있는 도로 세그먼트가  $B_2, B_3, B_4$  등 3가지이고, 유턴(U-Turn)과 같이 자기 자리를 돌아오는 경우는 배제된다고 가정하자(즉,  $N_{i,i}=0$ ).

$N_1$ 에서 발생한 모든 확률의 합은 1이다(즉,  $P(N_1)=1$ ). 그러므로,  $P(N_1) = P(N_{2,1}) + P(N_{3,1}) + P(N_{4,1}) = 1$  이므로  $P(N_{2,1}) = P(N_{3,1}) = P(N_{4,1}) = 1/3$ 이다. 이때,  $N_i$ 은 방문 횟수이므로 정수로 표현하기 위해  $N_1$ 을 3으로 놓게 되면,  $N_{2,1} = N_{3,1} = N_{4,1} = 1$ 이 된다. 그러므로, 확률  $P(B_2|B_1) = P(N_{2,1}|N_1) = 1/3$ 이다. 같은 방식으로  $P(B_3|B_1) = P(B_4|B_1) = 1/3$ 이다. 그러므로 한 번도 지나지 않은 셀 경계점  $B_1$ 에서 3개의 다른 경계점으로 진출할 확률은 33.33%임을 알 수 있다.

그림 2는 도로 네트워크를 이용하여 PCT-Tree가 셀 분할된 모습을 보여준다. 여기서  $C_i(i=1,2)$ 는 셀을 나타내며,  $B_j(j=1,4)$ 는 경계점, 그리고  $E_k(k=1,7)$ 는 셀의 진입 경계점과 진출 경계점을 연결한 에지(Edge)를 나타낸다. 여기에 과거 궤적  $E_1 \rightarrow E_3 \rightarrow E_5$ 를 20번 반복 후 각 셀의 셀 확률 매트



<그림 2> 셀 분할된 모습

$$CPM(C_1, I_k) = \begin{pmatrix} 0 & 1/23 & 21/23 & 1/23 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \end{pmatrix} \quad (\text{식 3})$$

$$CPM(C_2, I_k) = \begin{pmatrix} 0 & 1/23 & 1/23 & 21/23 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \end{pmatrix}$$

릭스를 살펴보자.

셀 C1과 C2의 셀 확률 매트릭스는 식 3과 같다.

이때, E1→E3→E5로 가는 확률 P(E1, E3, E5)를 구해 보자. 식 4는 셀 궤적의 확률을 구하기 위해 사용되는 Bayesian 이론이다.

$$P(b, a) = P(b|a) \cdot P(a) \quad (\text{식 4})$$

식 4에서 P(b, a)는 b와 a가 동시에 발생할 확률이며, P(b|a)는 a가 이미 발생한 후 b가 발생할 확률이다. 현재 이동 객체가 도로 세그먼트 E1에서 셀 C1으로 진입한다면, 확률은 P(E1)은 1이다. 그러므로 P(E3, E1) = P(E3|E1) · P(E1) = P(B3|B1) · 1 = 21/23 = 0.9130이다. 같은 방식을 적용하여 P(E2, E1) = P(E4, E1) = 1/23 = 0.0435이다. 여기서, E1과 E5는 확률 공간상 독립적이므로 P(E5|E1, E3) = P(E5|E3)이다. 그러므로, P(E1, E3, E5) = P(E1) · P(E3|E1) · P(E5|E1, E3)이므로, P(E1, E3, E5) = P(E1) · P(E3|E1) · P(E5|E3) = 1 · 21/23 · 21/23 = 0.8336이다. 비슷하게, P(E1, E3, E6) = P(E1, E3, E7) = 1 · 21/23 · 1/23 = 0.0397의 확률을 가진다.

진입하지 않은 경계점에 대해서는 진출할 확률이 모두 같으므로 확률 매개변수를 가질 필요가 없기 때문에 셀 확률 매트릭스 생성 시에 확률 변수는 객체가 진입한 경계점에만 생성된다. 또한, CT-

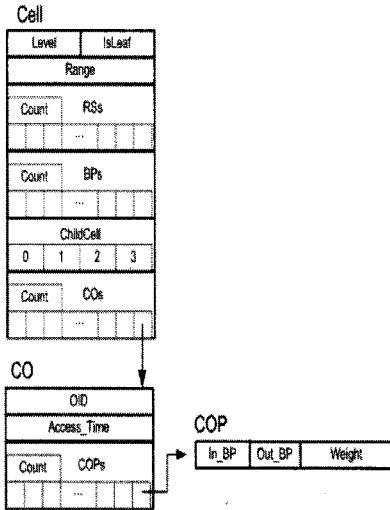
Tree는 장기간의 미래 궤적 예측을 위해 확률 검색 트리를 구성하지 않고 셀 기반의 확률 매트릭스를 운영하며, 미래 궤적 예측 시에 셀 확률 변수를 순회한다. 이때, 순회 규칙은 1, 2순위 확률을 순회하여 가장 확률 높은 미래 궤적을 검색한다. 이동 객체의 미래는 과거의 경험으로 얻은 확률로 100% 정확한 미래 위치를 예측할 수는 없기 때문에 불확실성이 내포된다. 그렇기 때문에 1, 2순위 순회로 인한 약간의 예측 오류(2.2%)를 통해 PCT\_Tree가 얻을 수 있는 장기간 미래 궤적 예측 속도의 향상은 미래 인덱스 전체 비용 면에서 매우 크다고 할 수 있다.

### 3.4 PCT-Tree의 알고리즘

본 절에서는 PCT-Tree의 자료 구조와 PCT-Tree를 위한 검색, 삽입, 삭제 알고리즘에 대해 설명한다.

#### 3.4.1 PCT-Tree 자료 구조

그림 3은 PCT-Tree의 자료구조를 보여 주는데, PCT-Tree는 크게 Cell, CO, COP의 구조체로 구성된다. Cell은 PCT-Tree의 셀 구조체이고, CO는 셀 객체의 구조체이고, COP는 셀 객체마다 가지는 진입, 진출 수를 저장한다. Cell 구조체에는



<그림 3> PCT-Tree 자료 구조

```

Function PCT_Search( Level, C, OID, In_BP, Prob, CT )
Input  Level: Prediction Level, C: Cell, OID: Object ID,
       In_BP: Enter BP, Prob: Probability, CT: Cell Trajectory
Output Cell Trajectory
Define Static Variable Cand_Prob_CTs //Cell Trajectory List
with probability

Begin
1. If Level is zero then Cand_Prob_CTs initialize
2. If Level > Level_Limit then //Level_Limit is prediction
   boundary level limit
   Add CT to Cand_Prob_CTs and Return
3. F_COP = Find COP that have the biggest weight among
   COPs with C, OID and In_BP
4. S_COP = Find COP that have the biggest weight except
   F_COP among COPs with C, OID and In_BP
5. Tot_W = Finds COPs' total weight that have given C, OID,
   and In_BP
6. F_CT = CT and Add CTS(C, In_BP, F_COP's next BP) to
   F_CT
7. PCT_Search( Level+1, F_COP's out CELL, OID, F_COP's
   out BP, Prob * F_COP's Weight / Tot_W, F_CT)
8. S_CT = CT and Add CTS(C, In_BP, S_COP's next BP) to
   S_CT
9. PCT_Search( Level+1, S_COP's out CELL, OID, S_COP's
   out BP, Prob * S_COP's Weight / Tot_W, S_CT)
10. CT = Find Cell Trajectory that have the biggest probability
    value in Cand_Prob_CTs
11. Return CT
End
    
```

<그림 4> 검색 알고리즘

셀 깊이(Depth)를 저장하는 Level, 단말 여부를 기록하는 IsLeaf, 셀 영역을 나타내는 Range, 도로 세그먼트의 리스트인 RSs, 경계점의 리스트인 BPs, 셀의 자식 셀을 가리키는 ChildCell, 그리고 셀 내부의 이동 객체의 리스트인 COs를 가진다. CO 구조체에는 객체 번호인 OID, 셀에 객체가 접근한 시각인 Access\_Time, 그리고 COP의 리스트인 COPs를 가진다. COP는 셀 확률 변수로서 진입 경계점 In\_BP, 진출 경계점 Out\_BP, 그리고 빈도수인 Weight를 가진다.

3.4.2 PCT-Tree 알고리즘

그림 4는 PCT-Tree의 미래 궤적 검색 알고리즘을 보여준다. 그림 4의 미래 궤적 검색 알고리즘을 살펴보면 다음과 같다. 라인 1에서 Level이 0이면 셀 궤적과 궤적의 확률을 가진 Candidate\_Prob\_CT를 초기화하고, 라인 2에서 Level이 검색의 한계치인 Level\_Limit보다 크면 후보 셀 궤적 CT를 Candidate\_Prob\_CT에 저장하고 알고리즘을 종료한다. 라인 3에서는 셀 객체 리스트 COPs에서 셀 C, 객체 식별자 OID, 진입 경계점 In\_BP를 가지며 가중치 Weight가 가장 큰 셀 객

체 확률 F\_COP를 구한다. 라인 4에서는 같은 방식으로 COPs에서 C, OID, In\_BP를 가지며 두 번째로 가중치 Weight가 큰 셀 객체 확률 S\_COP를 구한다. 라인 5에서는 COPs 중에서 C, OID, In\_BP를 가지는 모든 COP의 weight의 총합을 구하고, 라인 6에서는 입력된 셀 궤적 CT를 F\_CT에 대입한 후 C, In\_BP, F\_COP의 진출 경계점을 이용하여 셀 궤적 세그먼트 CTS를 만든 후 CTS를 F\_CT에 추가한다. 라인 7에서는 매개 변수 Level+1, F\_COP의 진출 셀, OID, F\_COP의 진출 경계점, 입력된 확률 Prob에 F\_COP의 확률을 적용한 새로운 확률, 그리고 새롭게 추가된 셀 궤적 F\_CT를 가지고 PCT\_Search 알고리즘의 재귀 호출을 수행한다. 라인 8에서는 입력된 CT를 S\_CT에 대입한 후 S\_COP에 대한 셀 궤적 세그먼트를 생성한 후 S\_CT에 추가한다. 라인 9에서는 매개변수 Level+1, S\_COP의 진출 Cell, OID, S\_COP의 진출 경계점, 입력된 확률 Prob에 S\_COP의 Weight의 비율을 적용한 확률, 그리고 S\_CT를 가지고 PCT\_Search의 재귀 호출을 수행한다. 라인 10에서는 재귀 호출의 결과인 Candidate\_Prob\_CT 중에서 가장 확률이 높은 셀 궤적을 구하는데, 이 셀 궤적이 미래 궤적이 된다. 마지막으



로 라인 11에서는 미래 궤적을 반환한다.

그림 5는 입력된 궤적을 PCT-Tree에 삽입하는 알고리즘을 보여준다. 그림 5의 삽입 알고리즘에서 과거 궤적 Traj가 입력되면, 라인 1에서 셀 궤적 CT로 변환된다. CT는 셀 궤적 세그먼트의 리스트이며, 셀 궤적 세그먼트에는 셀 ID, 객체의 진입 경계점, 그리고 진출 경계점을 가지고 있다. 라인 2에서는 CT안의 셀 궤적 세그먼트 CTS를 모두 조사하고, 라인 3에서 CTS의 셀에서 OID를 갖는 셀 객체 CO를 찾는다. 라인 4에서 CTS의 셀에 CO가 존재하지 않으면 새로운 CO를 생성하고, 라인 5에서 CO에 접근 시간을 저장한다. 라인 6에서 CTS의 진입 경계점과 진출 경계점을 가지는 셀 객체 확률 변수 COP를 CO 내에서 찾는다. 라인 7에서 COP가 없으면 새롭게 생성하고, 라인 8에서는 COP의 Weight를 1 증가시킨다.

그림 6은 PCT-Tree의 삭제 알고리즘을 보여준다. 그림 6의 PCT-Tree의 삭제 알고리즘을 살펴보면 먼저 라인 1에서 입력된 궤적이 셀 궤적 CT로 변환된다. 라인 2에서 CT의 모든 셀을 조사하고, 라인 3에서 CTS의 셀에서 OID를 갖는 CO를 찾는다. 라인 4에서 CTS의 셀에서 OID를 가지는 CO가 존재하지 않으면 다음 CTS를 조사하고, 라인 5에서 CTS의 진입 경계점과 진출 경계점을 가지는 COP를 CO 내에서 찾는다. 마지막으로 라인

6에서 COP가 존재하면 COP의 Weight 값을 1 감소시킨다.

#### 4. 성능평가

본 장에서는 본 논문에서 제시한 PCT-Tree의 성능평가를 살펴본다. 성능평가를 위해 사용된 시스템의 하드웨어 사양은 듀얼 Intel CPU 1.70GHz, 1GB RAM이며, 운영체제는 Windows XP를 사용하였다. 성능 평가에 사용된 테스트 데이터는 20만 개 도로 세그먼트 데이터를 이용하였다.

교차점 단위 확률 변수를 가지고 미래 궤적을 예측하는 PLM에서는 미래 궤적 예측 순회 시 100% 손실 없이 모든 경우의 궤적을 다 조사하기 때문에 장기간의 미래 궤적 예측이 매우 어렵다. 그러므로, 본 논문에서는 PLM 기법으로 순회를 하되 확률이 낮은 궤적에 대해서는 더 이상 순회를 하지 않는 PLM with Loss도 고려하였다. PLM with Loss에서 Loss값은 식 5를 통해 구해진다.

$$Loss = \left( \frac{1}{\text{교차점당 평균 도로 세그먼트 수} - 1} \right)^{(n+1)}, n \text{은 미래 궤적 탐색 깊이}$$

if Loss < 1/1000 then Loss = 1/1000 (식 5)

본 논문의 성능 평가에서는 PLM, PLM with

```

Algorithm PCT_Insert( OID, Traj )
Input  OID:Moving Object ID, Traj : Trajectory

Begin
1. Change trajectory Traj to cell trajectory CT
2. For each Cell Trajectory Segment CTS in CT do
   //Retrive all cell trajectory segment in cell trajectory CT gradually.
3. Find cell object CO with OID at CTS's cell
4. If there is no cell object that have OID at CTS's cell then Create new cell object CO
5. Input access time in cell object CO
6. Find cell trajectory variable COP that have CTS's in_BP and out_BP in cell object CO
7. If there is no cell object probability variable then Create new cell object probability variable COP
   Increment COP's Weight by one
8. End
    
```

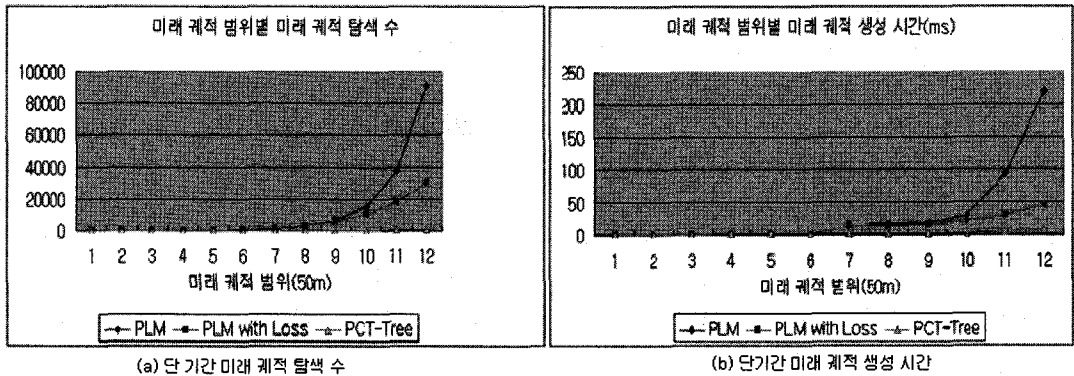
<그림 5> 삽입 알고리즘

```

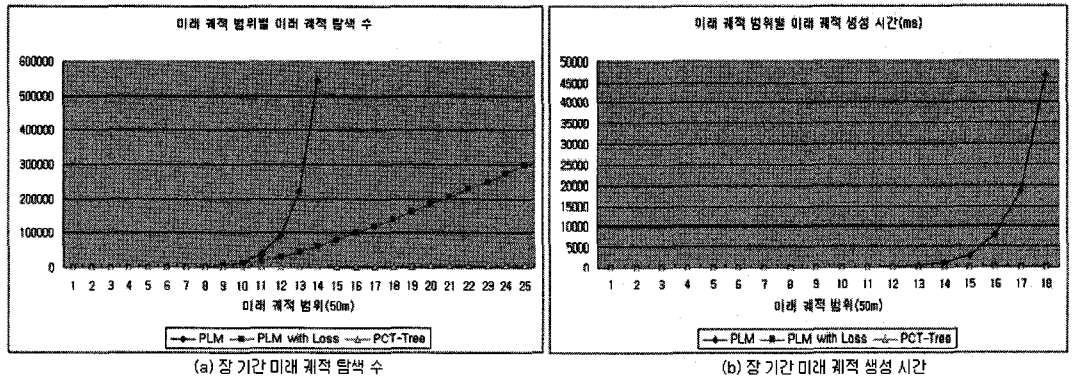
Algorithm PCT_Delete( OID, Traj )
Input  OID:Moving Object ID, Traj : Trajectory

Begin
1. Change trajectory Traj to cell trajectory CT
2. For each Cell Trajectory Segment CTS in CT do
   //Retrive all cell trajectory segment in cell trajectory CT gradually.
3. Find cell object CO that have OID at CTS's cell
4. If cell object CO does not exist in CTS's cell then goto 2
5. Find cell object probability variable COP that have CTS's in_BP and out_BP in cell object CO
6. If COP exists then
   Decrement COP's Weight by one
End
    
```

<그림 6> 삭제 알고리즘



<그림 7> 단기간 미래 궤적 범위별 미래 궤적 예측 성능 비교



<그림 8> 장기간 미래 궤적 범위별 미래 궤적 예측 성능 비교

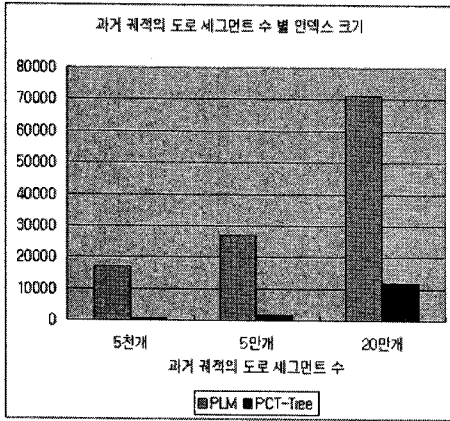
Loss, 그리고 PCT-Tree의 탐색 수와 미래 궤적 예측 시간을 비교하였다.

그림 7은 단기간 미래 궤적 범위별 미래 궤적 예측의 성능을 비교한 그래프이다. 본 논문에서 단기간이란 수 초에서 수 분에 이르는 시간을 의미하며, 거리상으로는 600m미만의 거리를 예측하는 경우이다.

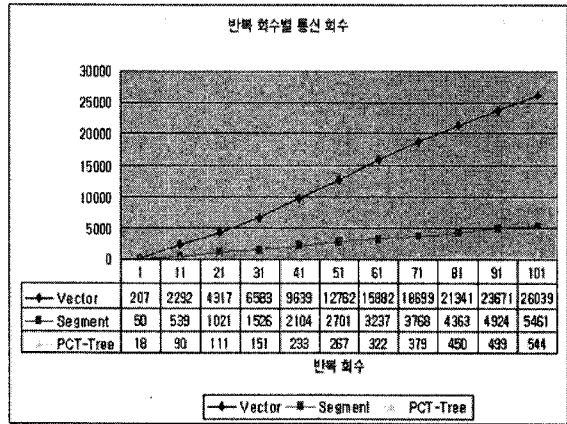
그림 7(a)과 그림 7(b)는 각각 단기간 미래 궤적 예측을 위한 미래 궤적 범위별 미래 궤적 탐색 수와 미래 궤적 범위별 미래 궤적 생성 시간을 보여준다. 그림 7(a)과 그림 7(b)에서 보듯이 주어진 미래 질의 시간 내 예측 가능한 미래 궤적 범위가 커질수록 미래 궤적 탐색 수 및 미래 궤적 생성 시간 모두에서 PCT-Tree가 PLM, PLM with Loss보다 더 우수해 지는 것을 알 수 있다.

그림 8은 장기간 미래 궤적 범위별 미래 궤적 예측의 성능을 비교한 그래프이다.

그림 8(a)과 그림 8(b)는 각각 장기간 미래 궤적 예측을 위한 미래 궤적 범위별 미래 궤적 탐색 수와 미래 궤적 범위별 미래 궤적 생성 시간을 보여준다. 그림 8(a)과 그림 8(b)에서와 같이 PLM은 미래 궤적 범위가 일정 범위를 넘을 경우 미래 궤적의 탐색 수 및 생성 시간이 기하급수적으로 증가한다. PLM with Loss는 장기간 미래 궤적 예측 시 일정 범위를 지난 경우 탐색 수는 미래 궤적 범위에 비례하여 증가하나, 생성 시간은 PCT-Tree와 유사한 성능을 보여준다. 그러나 PCT-Tree는 장기간 미래 궤적 예측 시 탐색 수 및 생성 시간에서 단기간과 마찬가지로 거의 고정적인 그래프를 보여준다. 즉, PCT-Tree는 단기간 뿐만 아니라



<그림 9> 미래 인덱스 크기 비교



<그림 10> 반복 횟수 별 통신 횟수 비교

장기간에서도 미래 궤적 범위에 따른 탐색 수 및 생성 시간의 변화가 거의 없음을 알 수 있다.

그림 9는 과거 궤적의 도로 세그먼트 수가 5,000개, 50,000개, 200,000인 경우 각각에 대한 PLM과 PCT-Tree의 인덱스 크기를 비교한 그래프이다. 그림 9에서와 같이 PCT-Tree는 이동 객체가 과거 지나간 진입 경계점과 진출 경계점의 셀 확률 변수만을 이용하여 셀 확률 매트릭스를 구성하므로 교차점별 확률 매트릭스를 가지는 PLM에 비해 아주 적은 인덱스 크기를 가진다.

그림 10은 이동 객체가 출발지에서 목적지로 반복 이동 시 PCT-Tree와 기존의 미래 위치 예측 방법인 세그먼트 트래킹과 벡터 트래킹을 사용할 때 미래 위치 예측 오류로 인한 클라이언트와 서버 간의 통신 횟수를 비교하는 그래프를 보여준다. 그림 10에서와 같이 PCT-Tree는 벡터 트래킹과 세그먼트 트래킹에 비해 신뢰성있는 미래 궤적을 예측하여 클라이언트에 전송되는 것을 보여주며, 그 결과 서버와 클라이언트 간의 통신 횟수가 줄어들어 그만큼 통신 비용이 감소하는 것을 알 수 있다.

### 5. 결론

기존의 미래 인덱스는 미래의 불확실성으로 인해 미래 위치를 예측하기가 어렵고, 예측 시간이 장기

간이 될수록 미래 위치 예측의 신뢰성이 더욱 떨어지는 문제점을 가진다. 또한, 서버는 이동 객체의 현재 정보를 최신으로 유지하기 위해 클라이언트의 현재 위치 정보를 주기적으로 갱신해야 되는데, 이러한 정보 갱신은 많은 통신 비용을 발생시키고, 빈번한 인덱스 갱신을 해야 하는 문제점을 가진다. 이러한 서버의 인덱스 갱신 부하는 전체 인덱스 성능을 감소시키며, 이로 인해 미래 위치 검색의 속도를 저하시킨다. 이와 같은 문제점을 해결하기 위해 본 논문에서는 도로 네트워크 환경에서 효율적인 장기간 미래 위치 예측을 위한 셀 기반의 미래 인덱스인 PCT-Tree를 제시하였고, 또한 성능 평가를 통해 PCT-Tree가 기존 미래 인덱싱 기법보다 우수함을 입증하였다.

기존의 확률에 의한 예측 기법이 장기간의 미래 질의 시 방대한 미래 궤적 탐색 부하로 인해 예측 성능이 떨어지게 되는 문제점을 해결하기 위해 PCT-Tree는 셀 단위 확률 매트릭스 기법을 사용한다. 또한 이를 통해 인덱스 크기를 줄일 수 있었으며, 미래 인덱스에 적합한 확률 검색 트리 순회 방법을 사용하여 신속한 미래 궤적 예측과 효율적인 이동 객체의 미래 위치 예측을 가능하게 한다. 그러므로, 도로 네트워크 환경에서 장기간 미래 위치 질의가 가능한 PCT-Tree는 다양한 위치 기반 서비스에서 미래 위치 예측 신뢰성을 높이는 위치

예측 엔진으로의 사용이 기대되며, 응급 차량이나 서비스 차량의 빠른 경로 안내를 위한 미래 교통 관제 시스템에서 미래 경로를 예측하는 인덱스로의 사용도 기대된다.

### 참고문헌

1. Ding, Z., and Guting, R. H., "Managing Moving Objects on Dynamic Transportation Networks," Proc. of the 16th International Conference on Scientific and Statistical Database Management, 2004, pp.287-296.
2. Lee, E. J., Rye, K. H., and Nam, K. W., "Indexing for Efficient Managing Current and Past Trajectory of Moving Object," Proc. of the 6th Asia-Pacific Web Conference on Advanced Web Technologies and Applications, 2004, pp. 782-787.
3. Saltenis, S., and Jensen, C. S., "Indexing of Moving Objects for Location-Based Services," Proc. of the 18th International Conference on Data Engineering, 2002, pp.463-472.
4. Civilis, A., Jensen, C. S., and Pakalnis, S., "Techniques for Efficient Road-Network-Based Tracking of Moving Objects," IEEE Transactions on Knowledge and Data Engineering, Vol.17 No.5, 2005, pp.698-712.
5. Civilis, A., Jensen, C. S., Nenortaitė, J., and Pakalnis, S., "Efficient Tracking of Moving Objects with Precision Guarantees," Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004, pp.164-173.
6. Karimi, H. A., and Liu, X., "A Predictive Location Model for Location-Based Services," Proc. of the 11th ACM International Symposium on Advances in Geographic Information Systems, 2003, pp.126-133.
7. Ding, R., Meng, X., and Bai, Y., "Efficient Index Maintenance for Moving Objects with Future Trajectories," Proc. of the 8th International Conference on Database Systems for Advanced Applications, 2003, pp.183-192.
8. Wolfson, O., and Yin, H., "Accuracy and Resource Consumption in Tracking and Location Prediction," Proc. of the 8th International Symposium on Advances in Spatial and Temporal Databases, 2003, pp.325-343.
9. Gowrisankar, H., and Nittel, S., "Reducing Uncertainty in Location Prediction of Moving Objects in Road Networks," Proc. of the International Conference on Geographic Information Science, 2002, pp.46-47.
10. 장유정, 김동오, 홍동숙, 한기준, "이동 객체의 효율적인 저장과 검색을 위한 화일 기반 이동 객체 저장 컴포넌트의 개발," 한국정보과학회 학술발표 논문집, 제31권 제1호, 2004, pp.118-120.
11. Frenzos, E., "Indexing Objects Moving on Fixed Networks," Proc. of the 8th International Symposium on Spatial and Temporal Databases, 2003, pp.289-305.
12. 이강준, 이동 객체 위치 예측을 위한 효율적인 쿼리 인덱싱 기법, 건국대학교 대학원 박사 학위 논문, 2006.
13. 배태욱, 안경환, 홍봉희, "도로 네트워크 환경에서 불확실성을 반영한 이동체 색인 및 질의 처리 기법," 한국정보과학회 데이터베이스 연구, 제20권 제2호, 2004, pp.156-163.

**홍동숙**

1999년 건국대학교 컴퓨터공학과(공학사)  
2001년 건국대학교 대학원 컴퓨터공학과(공학석사)  
2000년~2003년 쌍용정보통신 모바일/GIS 기술팀  
2003년~현재 건국대학교 대학원 컴퓨터공학과 박사  
과정  
관심분야: 데이터베이스, GIS, 이동체 데이터베이스,  
유비쿼터스 컴퓨팅

**김동오**

2000년 건국대학교 컴퓨터공학과(공학사)  
2002년 건국대학교 대학원 컴퓨터공학과(공학석사)  
2006년 건국대학교 대학원 컴퓨터공학과(공학박사)  
2006년~현재 건국대학교 컴퓨터공학부 강의교수  
관심분야: MODBMS, LBS, 스트림 데이터베이스,  
유비쿼터스 센서 네트워크, GML

**이강준**

1995년 건국대학교 컴퓨터공학과(공학사)  
1997년 건국대학교 대학원 컴퓨터공학과(공학석사)  
2006년 건국대학교 대학원 컴퓨터공학과(공학박사)  
2001년~현재 (주)엔에스텍 연구소장  
관심분야: MODBMS, GIS, LBS, 데이터베이스

**한기준**

1979년 서울대학교 수학교육학과(이학사)  
1981년 한국과학기술원(KAIST) 전산학과(공학석사)  
1985년 한국과학기술원(KAIST) 전산학과(공학박사)  
1990년 Stanford 대학 전산학과 Visiting Scholar  
1985년~현재 건국대학교 컴퓨터공학부 교수  
2004년~현재 한국정보시스템감리사협회 회장  
관심분야: 공간 데이터베이스, GIS, LBS, 텔레매틱스,  
정보시스템 감리