



changing the velocity of objects, it puts a maximum speed of road network segments, Experimental results show that the proposed scheme improves the accuracy for predicting positions of moving objects than other schemes based on the TPR-tree.

**Key words** : Road Network, Uncertainty, Moving Object Indexing, Query Processing

## 1. 서론

최근 휴대폰, PDA, 노트북과 같은 모바일 장치와 무선 이동 통신 기술, GPS, RFID와 같은 위치 보고 장치 기술의 발달로 위치 기반 서비스(LBS : Location Based Service)에 대한 요구가 점점 증대되고 있다. 위치 기반 서비스는 이동 통신망이나 GPS 등을 통해 개인이나 차량의 위치를 파악하여 긴급구조, 교통정보와 같은 부가서비스를 제공하는 신산업 분야로서, 향후 IT 산업뿐만 아니라, 전자상거래, 교통, 환경, 의료, 행정 등 사회 전 분야의 산업에 파급효과를 가져올 것으로 기대되고 있다.

위치 기반 서비스를 위한 기반 기술들로 크게 위치를 결정하기 위한 위치 측위 기술, 결정된 위치를 서버에 전송하기 위한 통신망 기술, 위치 데이터 관리를 위한 이동체 데이터베이스 기술 그리고 서비스를 제공하기 위한 응용 서비스 기술로 구성된다. LBS를 구현하는데 가장 필수적인 요소는 현재 이동체의 위치를 파악하는 위치 측위 기술로서 이동체의 위치를 추적하기 위해서 통신망의 기지국 위치를 활용하는 방식과 GPS 위성의 신호를 이용하는 방식으로 나눌 수 있다. 현재까지 주류를 차지해온 기술은 기지국 망을 이용하여 위치를 추적하는 셀 방식으로 기지국의 설계 방식 및 주변 환경에 따라 상당한 오차를 보여왔다. 최근에는 이보다 정확성이 뛰어난 GPS 위성을 이용하는 방식으로 변화하고 있는 추세이다. 위치 보고를 위해서는 이와 같은 위치 측위 기술을 사용하여 이동체의 위치를 결정하여야 한다.

그림 1은 LBS 기반 기술들을 이용하여 차량 관제 서비스를 제공하기 위한 일반적인 구성을 보여주고 있다. 구성을 살펴보면, 먼저 차량 안에 GPS 수신기, 무선 모듈, PDA를 결합한 단말기나 텔레메틱스(telematics) 전용 단말기가 GPS 위성을 이용하여 차량의 현재 위치 정보를 획득한 다음, 무선 통신을 이용하여 관제 서버에 주기적 또는 비주기적으로 위치를 보고 하게 된다. 관제 서버는 측위 시스템으로부터 획득된 차량 즉 이동체들의 위치를 저장, 관리하고 관리자는 관제 클라이언트를 이용하여 차량의 위치 조회나 차량에 지시를 할 수 있게 된다. 관제 서버는 기존의 데이터베이스가 지원하지 않는 위치 기반 질의를 지원해야 하고, 많은 이동체의 위치 보고를 효율적으로 저장, 검색하기 위해 이동체 데이터베이스를 사용하게 된다[1].

이동체 데이터베이스에서 말하는 이동체란 시간에 따라 연속적으로 위치 데이터가 갱신되는 객체를 말한다[2,3]. 이동체는 이동 점(moving point)과 이동 영역(moving region)으로 나눌 수 있다. 이동 점은 위치 정보만이 갱신되지만, 이동 영역은 시간에 따라 위치와 모양이 변하는 객체이다. 예를 들면 이동 점은 자동차, 동물, 비행기 등이 있고, 이동 영역은 폭풍의 영향권, 암세포의 상태 등이다.

이동체 데이터베이스의 질의 종류는 크게 영역 질의, 타임스탬프(timestamp) 질의, 궤적 질의, 복합 질의로 나누어진다[4,5]. 영역 질의는 주어진 시간 간격 동안에 공간 윈도우(spatial windows)에 속하는 이동체들을 검색하는 질의이며, 타임스탬프 질의는 특정 시간에 주어진 공간 윈도우에 속하는 이동체들을 검색하는 질의이다. 그리고 궤적 질의는 특정한 이동체의 궤적을 검색하는 질의이고, 복합 질의는 "특정 시간에 주어진 영역을 지나간 객체의 궤적을 검색하라"와 같이 시공간 도메인의 영역 질의와 궤적 질의가 복합된 질의이다. 영역질의는 다시 질의 시간에 따라 이동체의 과거 위치 또는 미래 위치에 대한 질의로 나눌 수 있다. 이 논문에서는 이동체의 현재 및 미래 위치 예측을 위한 영역 질의 처리 기법을 기술한다. 예를 들어 "5분 안에 시창 앞을 지나갈 수 있는 택시는?", "현재 시창 앞에 있는 택시는?"과 같은 질의 처리를 대상으로 한다.

대표적인 현재 및 미래 위치 색인으로 다차원 색인 R\*-tree[6]를 기반으로 하는 TPR-tree[7]가 있다. TPR-tree는 시간을 선형함수에 대한 매개 변수로 사용하여, 연속적으로 이동하는 이동체의 갱신비용을 줄이면서 현재 및 미래 위치 검색을 지원하는 색인이다. 그러나 시간에 대해 선형적으로 이동체의 위치를 예측하기 위해서는 이동체의 이동 방향과 속도가 특정한 임계값을 넘어서는 경우에 새로운 위치 정보를 보고 해야만 한다. 예를 들어, 이동체가 마지막으로 보고한 속도와 5km 이상 차이가 날 경우에 클라이언트인 이동체는 서버로 현재 위치 정보를 다시 보고한다. 그러나 이러한 조건은 차량과 같이 교통 상황에 따라 속도와 방향이 빈번하게 변화는 환경에 적용할 경우 이동체의 잦은 위치 보고로 클라이언트인 이동체와 서버와의 무선 통신 비용을 크게 증가시키는 문제가 있다. 그리고 이동체의 수가 많아지면 다량의 보고에 따른 서버의 병목 현상을 초래할 수 있는 문제가 있다.

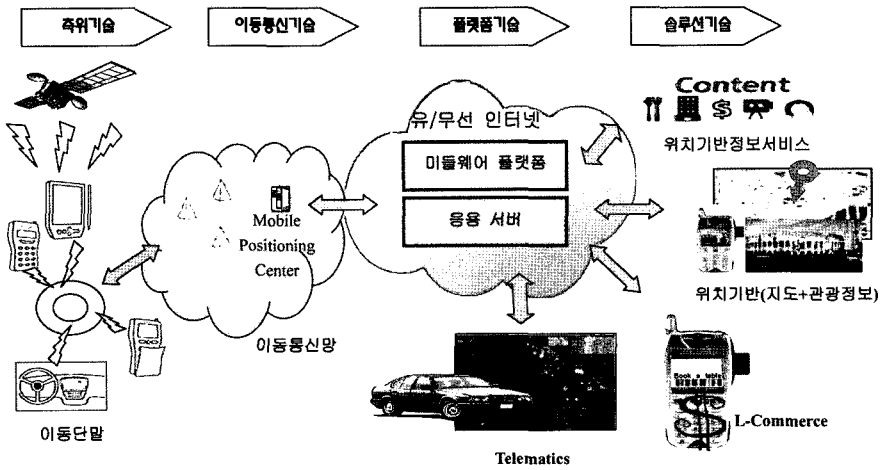


그림 1 위치 기반 서비스 개요

적정한 통신 비용을 유지하기 위해서는 일정한 시간 간격에 따라 주기적인 위치 보고를 수행하는 방법이 있다. 주기적인 위치 보고를 수행하면 비주기적인 위치 보고와 같이 이동체의 속도와 방향이 변화해도 서버로의 새로운 위치 보고를 하지 않기 때문에 보고 횟수를 크게 줄일 수 있다. 그러나 보고 주기 사이에 이동체의 속도와 방향의 변화로 인해 TPR-tree와 같이 시간에 대해 선형적으로 이동체의 위치를 예측하는 색인의 경우 질의 영역이 주어졌을 때 색인에서의 질의 결과는 이동체의 마지막 보고 된 속도와 방향 정보를 사용하기 때문에 실제와는 다른 위치를 예측하게 되어 질의 영역에 포함되지 않는 이동체이지만 질의 결과를 반환하는 문제가 있다. 또한, 이동체의 마지막 위치 보고 이후 속도와 방향이 변화하여 질의 영역에 포함되지만 결과로 검색되지 못하는 문제도 발생할 수 있다[8,9].

이 논문에서는 주기적인 위치 보고 환경에서 보고 주기 사이에 이동체의 이동 속도와 방향의 변화로 인한 위치 예측 오차가 발생하는 문제점을 해결하기 위하여 도로 네트워크를 이용한 불확실 위치데이터의 처리 방법을 제안한다. 또한, 도로 네트워크 및 이동체의 위치를 저장하기 위해서 R-tree를 이용한 데이터 저장 구조를 제시하며 이를 이용하여 이동체의 위치를 예측하기 위한 영역 질의 처리 기법을 제시한다. 제시된 기법은 도로 네트워크 정보를 이용함으로써 이동체의 이동 방향을 도로 네트워크 세그먼트의 방향으로 제한하여 불확실 영역을 감소시키고 있으며 도로 네트워크 세그먼트의 종류별로 최대 속도를 설정하여 이동 속도의 변화에 대한 불확실성을 제거한다. 실험결과를 통하여 제안된 질의 처리 기법이 미래 위치에 대한 영역 질의 시에 질의 영역에 포함되지만 결과로 검색되지 못하는 문제

를 발생시키지 않으면서 질의 영역에 포함되지 않는 이동체의 검색을 최소화 함으로써 TPR-tree를 이용한 기존의 질의 처리 기법보다 위치 예측 정확도가 향상됨을 확인할 수 있다.

이 논문의 구성은 다음과 같다. 2장에서 관련 연구 기술을 기술하고, 3장에서는 일정한 시간 간격마다 보고에 따른 이동체의 불확실 위치데이터에 대한 문제 정의를 기술한다. 4장에서는 3장의 문제점을 해결할 수 있는 도로 네트워크를 이용한 질의 처리 기법을 제안하고, 5장에서는 질의 처리 알고리즘을 제시한다. 그리고 6장에서는 실험을 통하여 영역 질의에 대한 성능평가를 보인다. 마지막으로 7장에서는 결론 및 향후 연구를 기술한다.

## 2. 관련 연구

이 장에서는 이동체 모델링에 대한 연구를 먼저 기술하고, 다음으로 이동체의 현재 및 미래 위치 색인에 대한 연구와 이동체의 불확실 위치데이터에 대한 연구에 대해 기술한다.

### 2.1 이동체 모델링에 대한 연구

이동체 모델링은 크게 과거 위치 데이터와 미래 위치 데이터에 대한 모델링으로 구분된다. 과거 위치 데이터는 이동체가 이동한 궤적 선분(trajecory segment)과 MBB로 이동체를 표현하며, 미래 위치 데이터는 이동체의 좌표 정보와 속도 정보를 이용하여 이동체의 보고 시간을 매개변수로 하는 선형함수로 표현하는 방법이 대표적이다.

이동체의 과거 위치 데이터 모델링 방법으로는 궤적 선분을 사용하여 이동체의 위치를 시공간 상의 점으로 표현하고, 연속적인 두 위치를 선형 보간법(linear interpolation)을 사용하여 3차원 선분으로 표현하는 방법이

대표적으로 사용된다[4,5]. 이와 같은 과거 위치 모델링 방법은 대부분 R-tree[10]를 기본 구조로 사용하기 때문에 객체를 최소 경계 박스(MBB : Minimum Bounding Box)의 형태로 표현하며 일반적으로 객체 선분은 3차원의 MBB로 색인에 저장된다.

이동체의 미래 위치 데이터를 모델링하는 대표적인 방법으로 이동체의 위치를 선형적인 시간 함수를 사용하여 표현하는 방법[7]이 있다. 그림 2의 (a)와 같이 측위 시스템으로부터 얻을 수 있는 이동체의 위치 좌표와 보고 시간, 그리고 속도 정보를 이용하여 시간을 매개변수로 하는 선형함수로 표현하기 때문에 이동체의 위치 정보 갱신 횟수를 최소화하는 장점이 있으며, 이와 같은 방법은 현재 및 미래 위치 질의를 위한 색인에서 많이 사용된다. 그러나 그림 2의 (b)와 같이 이러한 선형적인 예측의 정확성을 보장하기 위해서는 이동체의 이동 방향이나 이동 속도가 미리 설정된 임계값 이상을 벗어날 경우에 새로운 위치 정보로 갱신해 주어야만 설정된 오차 범위 내에서 예측의 정확성이 보장된다. 따라서, 주기적인 위치 보고 환경에서는 선형적인 위치 예측 방법만으로는 정확성을 보장할 수 없다. 이 논문에서는 주기적인 위치 보고 환경에서 이동체의 미래 위치 예측의 정확성을 보장하기 위해서 부가적으로 도로네트워크를 사용한 질의 처리 방법을 제안한다.

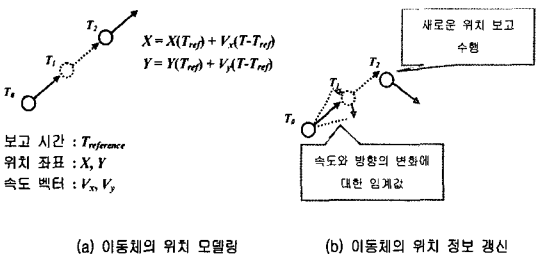


그림 2 선형함수를 사용한 이동체의 미래 위치 모델링

2.2 이동체 색인에 대한 연구

이동체 색인에 관련된 연구는 크게 과거 위치 색인과 미래 위치 색인에 대한 연구로 나눌 수 있다. 이 논문에서는 이동체의 미래 위치에 대한 질의 처리 기법에 대한 연구이므로 대표적인 미래 위치 색인인 TPR-tree[7]와 이의 기반이 되는 R-tree[10]와 R\*-tree[6]에 대해 기술한다. TPR-tree가 발표된 이후 이동체의 현재 및 미래 위치 색인에 대한 연구[11-13]들이 활발히 진행되었다. 그러나 TPR-tree를 기반으로 성능 향상을 초점을 두고 있으며, TPR-tree와 마찬가지로 시간에 대해 선형적인 예측방법을 사용하기 때문에 주기적인 위치 보고 환경에 적용할 경우 예측 오차가 발생하는 문제가 있다.

따라서, 미래 위치 색인에 대한 관련 연구로는 TPR-tree에 대해서만 기술한다.

2.2.1 R-tree와 R\*-tree

R-tree[10]는 데이터 분할 방법의 대표적인 공간 색인으로서 공간 객체를 최소경계사각형(MBR : Minimum Bounding Rectangle)을 사용하여 표현하고, 저장하는 균형 트리 구조의 색인이다. 공간 객체를 색인에 삽입 시에 사용되는 ChooseSubtree 알고리즘은 최소 영역 확장 정책(Least Area Enlargement Policy)를 사용하여 삽입할 노드를 선택한다. 또한 노드 오버플로우가 발생했을 시에는 분할되는 2개의 노드가 최소 영역을 갖도록 분할한다. 이러한 이유로 R-tree의 삽입, 분할 정책은 최소 영역 정책이라 할 수 있다.

R\*-tree[6]는 다차원의 정적 데이터를 효율적으로 색인하기 위하여 연구되었다. 그림 3은 2차원에서의 R\*-tree 구조를 보여주고 있다. 10개의 사각형(a, b, ..., j)들이 공간 근접성을 고려하여 클러스터링 되어있으며, 4개의 단말 노드 N1, ..., N4와 이를 포함하는 N5, N6을 엔트리로 포함하는 루트 노드로 구성되어 있다. 각 엔트리는 최소 경계 사각형으로 표현된다. 단말 노드 엔트리의 최소 경계 사각형은 실제 객체를 포함하고 비단말 노드 엔트리의 최소 경계 사각형은 자식 노드들을 포함하는 구조이다. R\*-tree는 R-tree의 단점인 노드간의 심한 중첩(overlap)과 사각 공간(dead space)으로 인한 성능 저하를 개선하기 위하여 영역 크기, 각 최소 경계 사각형의 둘레, 같은 노드에 위치하는 두 경계 사각형 간의 중첩 등을 모두 고려하는 새로운 분할 알고리즘과 재삽입 알고리즘을 사용하여 영역 질의에서 뛰어난 성능을 보여준다.

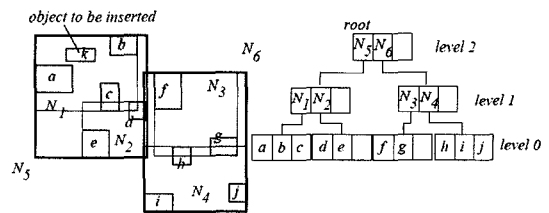


그림 3 R\*-tree의 구조

2.2.2 TPR-tree

TPR-tree[7]는 다차원 색인인 R\*-tree기반의 색인으로 이동체의 현재 및 미래 위치 질의를 지원하기 위하여 다차원 색인인 R-tree의 최소 경계 사각형 대신에 비단말 노드의 엔트리 정보로 시간 함수 기반의 경계 사각형(BR : Bounding Rectangle)과 경계 사각형의 각 변에 대한 속도 벡터 그리고 참조 시간을 저장한다. 질의 시에 각 경계 사각형의 참조 시간과 질의 시간과의

차이를 속도 벡터에 곱하여 이동체의 미래 위치를 시간에 대해 선형적으로 예측함으로써 이동체의 위치 정보 보고에 대한 갱신 횟수를 최소화하였다. 그리고 이동체의 위치가 시간에 대해 선형적으로 예측된 위치 내에 있음을 보장하기 위하여 속도와 방향이 특정한 임계값을 벗어날 경우에는 새로운 위치를 보고한다.

경계 사각형은 그림 4와 같이 내부에 포함된 모든 엔트리를 항상 포함하기 위하여 각 방향으로 가장 빠른 속도로 이동하는 엔트리의 속도 값으로 경계 사각형의 확장 속도를 설정한다. 내부의 모든 엔트리들을 항상 포함하는 경계 사각형은  $\langle B_{left}, B_{right}, B_{top}, B_{bottom}, V_{left}, V_{right}, V_{top}, V_{bottom}, t_{reference} \rangle$ 으로 표현되며, 질의 시간  $t_q$ 가 주어졌을 때  $t_q$ 와  $t_c$ 의 차이만큼 경계 사각형을 선형적으로 확장시켜 질의를 처리한다.

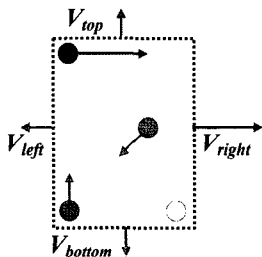


그림 4 Time-Parameterized Bounding Rectangle

그림 5의 (a)는 참조 시간이  $t_0$ 인 경계 사각형  $N_1, N_2$ 와 이에 포함된 엔트리  $a, b, c, d$ 를 보여주고 있다. 그림 5의 (b)는  $t_1$  시간에 경계 사각형들의 모습이다. 즉,  $t_1 - t_0$  시간만큼 각 경계 사각형의 속도 벡터에 곱하여 경계 사각형을 확장시키게 되고, 주어진 질의 영역과 확

장된 경계 사각형이 교차하게 되면 하위 노드를 계속해서 검색하게 된다. 그러나 TPR-tree는 시간에 대한 선형함수를 사용하고 이 선형적인 예측은 이동체의 속도와 방향이 변할 경우에 새로운 위치보고를 수행하여만 정확한 예측을 보장하기 때문에 이 논문에서 대상 환경으로 하는 일정한 시간 간격으로 위치 보고를 수행하는 이동체에 적용할 경우 예측 오차가 발생하는 문제가 있다.

### 2.3 불확실 위치데이터에 대한 연구

이동체의 위치 좌표나 속도의 측정 시 발생하는 오차 그리고 위치 정보의 샘플링 시 발생하는 이동체의 불확실 위치데이터에 관한 연구로 [14-16]이 있다. [14]는 이동체의 보고된 두 위치 사이의 존재할 수 있는 경로를 추정하는 연구이고, [15]는 이동체의 시공간 변화에 대해 확률과 퍼지이론을 이용한 모델링에 관한 연구이다. 그러나 [14,15]는 이동체의 과거 위치데이터에 대한 연구로 이 논문에서 대상으로 하는 이동체의 미래 위치에 대한 불확실성에 적용할 수 없는 문제가 있다. [16]은 그림 6과 같이 TPR-tree를 사용하여 이동체의 현재 및 미래 위치데이터에 대한 불확실성을 반영하는 방법으로 색인에 불확실성 정보를 기록하는 방법과 질의에 불확실성을 반영한 방법을 제안하였다. 색인에 불확실성을 적용하는 방법은 색인에 삽입 시에 이동체의 위치 정보와 발생 가능한 오차 정보를 함께 추가하여 색인함으로써 각 이동체의 특성에 따른 오차 정보를 설정하여 처리할 수 있는 장점이 있다. 그리고 질의에 불확실성을 반영하는 방법으로 발생할 수 있는 최대 오차 크기만큼 질의 영역을 상대적으로 확장시켜서 질의를 처리하는 방법이다.

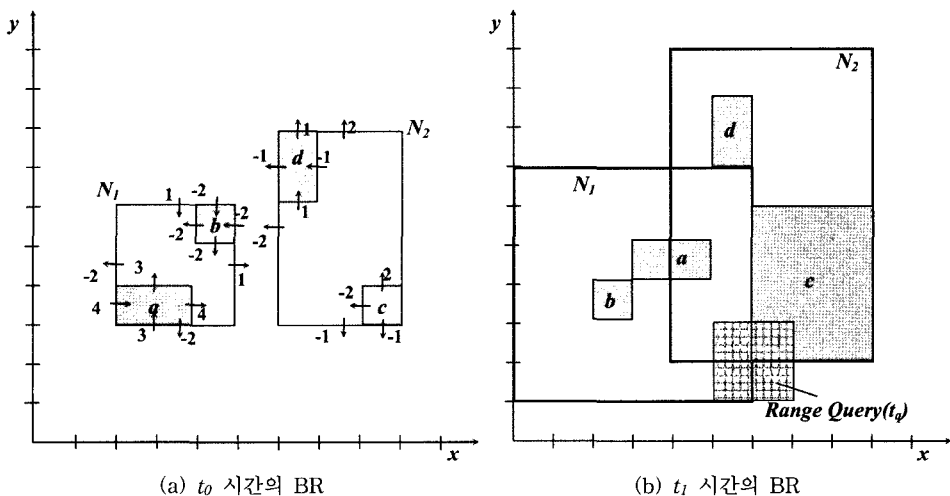


그림 5 TPR-tree의 시간에 따른 BR 확장 구조

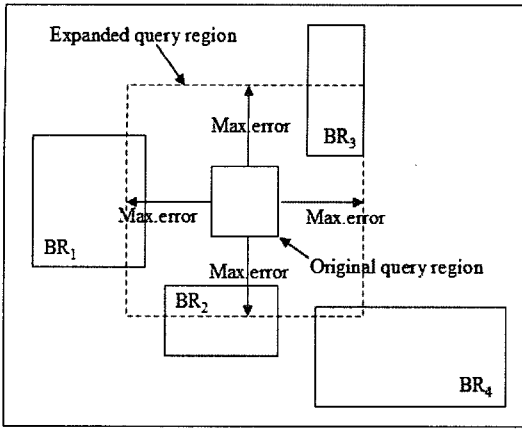


그림 6 질의 영역을 최대 오차 크기로 확장

그러나, [16]에서 제안한 방법들은 이동체의 위치에측 시 발생할 수 있는 오차 크기를 미리 설정해야 하는 단점이 있으며, 최대 오차를 이용하여 질의를 처리할 경우 실제 질의 조건을 만족하지 않지만 결과로 포함되는 객체의 수가 증가하는 문제가 있다.

3. 문제 정의

이 장에서는 비주기적인 위치 보고와 주기적인 보고의 차이점에 대해 기술하고, 주기적으로 위치 보고를 수행하는 이동체들의 미래 위치 예측에 대한 문제점을 기술한다.

이동체들은 서버에 위치를 보고할 때 주기적 또는 비주기적으로 보고를 한다. 비주기적인 보고는 TPR-tree에서와 같이 이동체의 이동 방향이나 속도가 미리 설정된 임계값을 넘어서게 되면 위치 보고를 수행하는 경우를 말한다. 그림 7의 (a)는 도로 네트워크를 따라 이동하는 차량들의 이동 속도와 방향의 변화에 따른 위치 보고를 나타내며, 이러한 속도와 방향의 변화에 따른 비

주기적인 위치 보고는 새로운 위치 갱신이 있기까지 동일한 속도와 방향으로 이동함을 보장할 수 있기 때문에, TPR-tree와 같이 이동체의 위치를 시간에 대해 선형적으로 예측하여 저장하는 것이 가능하다. 그러나, 차량과 같이 이동체의 이동 속도와 방향의 변화가 잦은 환경에 속한 이동체들의 경우에는 매우 빈번한 위치 보고를 필요로 하게 되어 통신 비용을 증가시키는 단점이 있다. 통신 비용의 증가를 막기 위하여 주기적인 위치 보고를 하게 되면 이동체의 위치 보고 횟수를 크게 줄일 수 있는 장점이 있으나 그림 7의 (b)와 같이 이동체의 속도와 방향이 변해도 위치 보고를 수행하지 않기 때문에, 이동체의 마지막 위치 보고 이후 미래 위치에 대한 정확한 예측을 할 수 없는 문제점이 있다[8,9].

그림 8은 주기적인 위치 보고를 수행하는 환경에서 이동체의 미래 위치 질의 처리 시 문제점을 나타낸다. TPR-tree와 같이 선형적으로 이동체의 위치를 예측하는 색인을 사용하여 질의를 처리할 경우에 질의 영역 Q<sub>I</sub>가 주어졌을 때 색인에서의 질의 결과는 이동체의 마

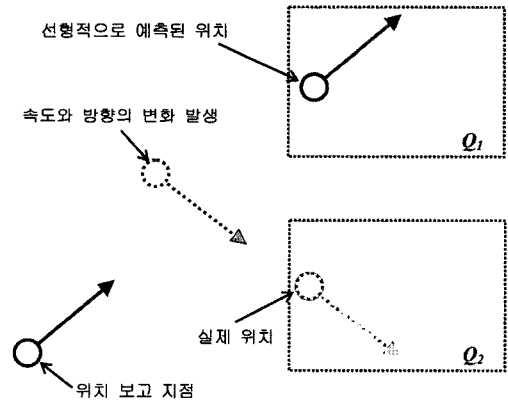
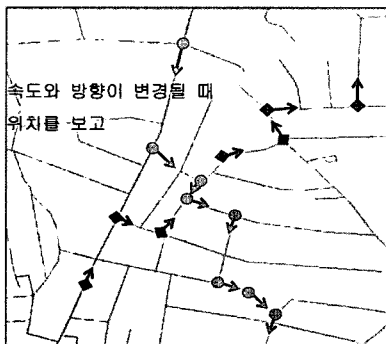
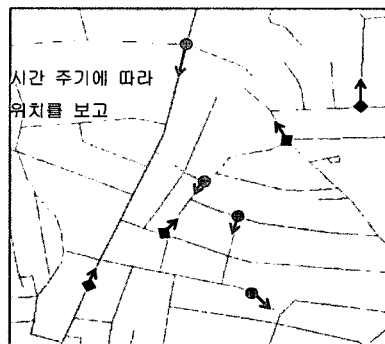


그림 8 주기적인 위치 보고 시 문제점



(a) 비주기적인 위치 보고



(b) 주기적인 위치 보고

그림 7 이동체의 위치 보고

지막 보고 된 속도와 방향 정보를 사용하기 때문에 실제와는 다른 위치를 예측하게 되어, 질의 영역  $Q_1$ 에 포함되지 않는 이동체이지만 이를 질의 결과로 반환하는 문제가 있다. 그리고 질의 영역  $Q_2$ 가 수행될 경우에는 이동체의 마지막 보고 이후에 속도와 방향이 변화하여 실제로 질의 영역  $Q_2$ 에 포함됨에도 불구하고 질의 결과로 검색되지 못하는 문제가 발생한다.

앞으로의 논문 전개를 위하여 위와 같은 상황에 대해서 표 1에서 용어 정의를 한다.

이 논문에서는 이러한 주기적인 위치 보고 환경에서 이동체의 미래 위치 예측 시에 문제가 되는 *False miss*의 발생을 피하고, *False hit*를 최소화하기 위한 도로 네트워크 기반 불확실 위치데이터의 처리 기법을 제안한다. 또한, 도로 네트워크 및 이동체의 위치를 저장하기 위해서 R-tree를 이용한 데이터 저장 구조를 제시하며 이를 이용하여 이동체의 위치를 예측하기 위한 영역 질의 처리 기법을 제시한다. 제시된 기법은 도로 네트워크 정보를 이용함으로써 이동체의 이동 방향을 도로 네트워크 세그먼트의 방향으로 제한하여 불확실 영역을 감소시키고 있으며 도로 네트워크 세그먼트의 종류별로 최대 속도를 설정하여 이동 속도의 변화에 대한 불확실성을 제거한다. 실험결과를 통하여 제안된 질의 처리 기법이 TPR-tree를 이용한 질의 처리 기법보다 이동체의 위치 예측 정확도가 향상됨을 확인할 수 있다.

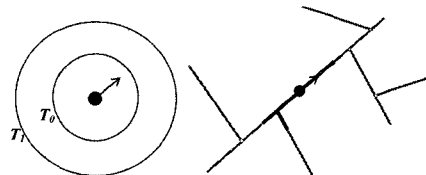
#### 4. 불확실 위치데이터의 질의 처리

이 장에서는 3장에서 언급한 주기적인 위치 보고 환경에서 이동체 미래 위치를 저장하는 색인을 이용한 질의 처리 시에 발생하는 문제점들을 해결하기 위하여 도로 네트워크를 이용한 질의 처리 방법을 제시한다. 우선 도로 네트워크를 사용하여 불확실 위치데이터를 처리하는 방법에 대해 기술한 후 제안하는 질의 처리 구조에 대해서 기술한다. 마지막으로, 제안한 방법과 구조를 사용하여 이동체의 미래 위치에 대한 영역 질의를 처리하는 방법에 대해 기술한다.

#### 4.1 도로 네트워크를 이용한 불확실성 처리

이 절에서는 도로 네트워크를 사용하여 이동체의 이동 방향과 속도의 변화에 대한 불확실성을 반영하는 방법을 제시한다.

첫째, 이동체 이동 방향의 변화에 대한 불확실성을 처리하기 위해서 도로 네트워크를 사용하는 방법을 제안한다. 주기적인 위치 보고 환경에서 이동체의 이동 방향에 대한 불확실성을 반영하기 위해서는 그림 9의 (a)와 같이 이동체가 이동할 수 있는 모든 방향을 고려해야 한다. 이는 보고 주기 간격이 커질수록 불확실 영역 또한 매우 커지기 때문에 *False hit* 수를 증가시키는 문제가 있다. 그러나 그림 9의 (b)와 같이 도로 네트워크 정보를 이용하여 이동체의 이동 방향을 도로 네트워크 세그먼트의 방향으로 제한할 경우 불확실 영역을 크게 감소시킬 수 있기 때문에 *False hit*를 줄일 수 있는 장점이 있다. 즉, 이동체가 마지막 보고한 위치의 네트워크 세그먼트에서부터 해당 세그먼트의 인접 리스트 정보를 사용하여 이동체의 이동 가능한 방향을 제한한다. 이 논문에서는 도로 네트워크상에 있는 이동체가 도로 네트워크 세그먼트의 모든 방향으로 이동이 가능하다고 가정하고 있기 때문에 질의 처리 시에 도로 세그먼트의 모든 방향으로의 이동에 대해서 고려를 하고 있다. 또한, 여러 개의 분기점 네트워크가 검색되었을 경우에는 이동 가능한 모든 분기 경우를 고려하기 때문에 이동 방향의 변화에 대한 불확실성을 제거하고 있다. 즉, 이동체의 보고 주기 사이에 이동 방향의 변화로 인한 *False miss*를 방지할 수 있다.



(a) 유클리디언 공간에서의 불확실 영역 (b) 네트워크 공간에서의 불확실 영역  
그림 9 도로 네트워크를 이동하는 이동체

표 1 False hit/False Miss

False hit/False miss	
영역 질의를 통하여 이동체의 위치를 예측 할 경우에 실제로 질의 영역에 포함되지 않지만 질의 결과로 반환되는 경우 <i>False hit</i> 가 발생했다고 정의한다. 그리고, 실제로 질의 영역에 포함되지만 질의 결과로 검색되지 않는 경우 <i>False miss</i> 가 발생했다고 정의한다.	
검색결과 집합에 실제 질의 조건을 만족하는 이동체 수를 $H$ , 검색된 총 후보 객체 수를 $P$ , 실제 질의 조건을 만족하는 총 이동체 수를 $R$ 이라고 했을 때 <i>False hit</i> 와 <i>False miss</i> 는 다음과 같은 수식으로 정의될 수 있다.	
$False\ hit(\%) = \frac{P - H}{P} \times 100(\%)$	
$False\ miss(\%) = \frac{R - H}{R} \times 100(\%)$	

둘째, 이동체의 이동 속도의 변화에 대한 불확실성을 반영하기 위하여 그림 10과 같이 세그먼트 종류별로 이동체가 이동 가능한 최대 속도를 설정하는 방법을 제안한다. 예를 들어, 고속 도로와 일반 시내 도로, 1차선 도로, 4차선 도로와 같이 네트워크 세그먼트 별로 이동체의 이동 가능한 최고 속도를 제한하여 질의 처리 시에 이동체가 속한 세그먼트의 최대 속도로 이동체의 움직임을 예측하여 이동 속도의 변화로 인한 *False miss*를 방지한다. 즉, 이동체가 위치하는 해당 세그먼트의 최대 속도를 초과하지 않을 경우에 속도의 변화로 인한 *False miss*를 방지 할 수 있다.

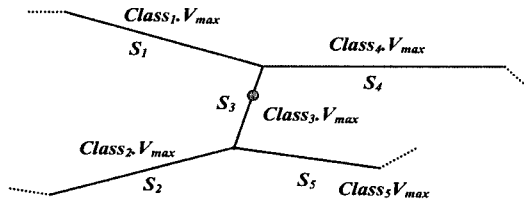


그림 10 네트워크 세그먼트 종류별 최대 속도 설정

4.2 도로 네트워크를 이용한 자료 구조

그림 11에서와 같이 질의 처리를 위한 구조는 네트워크 세그먼트 공간 정보를 위한 2차원 R-tree와 이동체의 위치 정보를 위한 2차원 R-tree로 구성된다. 네트워크 공간 정보를 위한 R-tree는 영역 질의 시에 질의 영역에 포함된 네트워크 세그먼트를 검색하거나 이동체 속한 네트워크 세그먼트를 검색할 시에 사용된다. 이동체의 미래 위치를 검색하기 위한 색인으로써 TPR-tree 대신에 R-tree를 사용한다. TPR-tree의 경우 그림 12의 (a)와 같이 경계 사각형 내에 존재하는 이동체 중에서 경계 사각형의 각 변으로 가장 빠른 속도로 이동하고 있는 이동체의 속도를 반영하는 동적인 경계 사각형 구조를 사용한다. 그러나 주기적으로 이동체의 위치 보기가 수행이 될 경우에는 이동 방향이나 속도가 변하게 되면 이동체가 항상 경계 사각형 내에 포함됨을 보장할 수 없는 문제가 발생한다. 따라서, 이 논문에서 사용하는 R-tree구조는 그림 12의 (b)와 같이 TPR-tree계열 색인과는 달리 비단말 노드의 엔트리에 속도 벡터 값을 저장하지 않고 이동체들을 포함하는 최소 경계 사각형과 보고 시간만을 저장한다. 즉, 단말 노드의 각 엔트리는 이동체의 위치 좌표와 참조 시간으로 구성되고 비단말 노드의 참조 시간은 자식 노드의 엔트리 중에서 가장 오래된 참조 시간을 저장하는 구조이다.

그림 13은 이 논문에서 사용하는 네트워크 세그먼트의 데이터 구조이다. *SID*는 네트워크 세그먼트의 식별자로서 질의 처리 시에 질의 영역에 포함되는 세그먼트

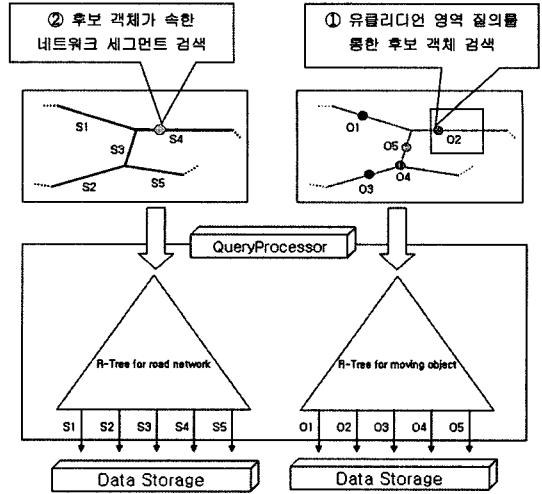
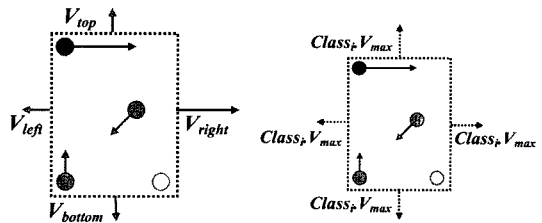


그림 11 도로 네트워크를 이용한 색인 구성 및 질의 처리 방법



(a) TPR-Tree 기반 BR 구조 (b) 불확실 위치반영 BR 구조

그림 12 TPR-tree와 불확실 위치데이터를 반영한 BR 구조 비교

<i>SID</i>	<i>S.Point</i>	<i>E.Point</i>	<i>Class</i>	<i>Adj.List</i>	<i>Distance</i>
------------	----------------	----------------	--------------	-----------------	-----------------

그림 13 네트워크 세그먼트 데이터 구조

들의 식별자를 저장하고, 이동체가 이동할 수 있는 세그먼트들의 식별자와 같은지 비교하기 위하여 사용된다. *S.Point*와 *E.Point*는 세그먼트의 시작점과 끝점으로 세그먼트의 기하정보이다. 이는 이동체가 위치하고 있는 세그먼트를 찾을 시에 사용된다. *Class*는 세그먼트 별로 이동체가 이동할 수 있는 최대 허용 속도를 구별하기 위한 정보이다. *Adj.List*는 세그먼트와 이웃하고 있는 세그먼트들의 인접 리스트 정보로써 세그먼트 식별자 리스트를 저장하고 있다. *Distance*는 해당 세그먼트의 총 길이를 나타낸다.

4.3 불확실 위치데이터에 대한 영역 질의

불확실 위치데이터에 대한 영역 질의 처리는 그림 11의 색인 구조를 사용하여 처리된다. 이동체 색인에서 유클리디언 영역 질의를 통해서 후보 객체를 검색한 후



반환된 결과에 대해서 도로 네트워크 색인 통하여 후보 객체가 속한 네트워크 세그먼트를 검색함으로써 영역에 포함된 이동체들을 결과로 반환한다. 질의 영역이  $QR$ , 질의 시간이  $QT$ , 마지막 위치 보고 시간을  $T_{ref}$ 라 할 때 이동체 색인의 비단말 노드의 엔트리인 MBR을 각 방향으로  $(QT - T_{ref})$  시간만큼 유클리디언 거리로 확장시킨다. 확장 속도는 그림 12의 (b)와 같이 가장 큰 최대 속도를 가지고 있는 네트워크 세그먼트 클래스의 속도로 확장시킨다. 따라서 이동 객체의 방향과 속도가 변하더라도 확장된 MBR안에 항상 포함된다. 그리고 확장된 MBR이  $QR$ 과 교차된다면 단말 노드에 도달할 때까지 같은 방식으로 해당 엔트리의 하위 노드들에 대해  $QR$ 과 교차하는지를 검사하고 단말 노드에 도달하면 해당 노드에 포함된 이동체들을 결과로 반환한다.

반환된 결과에 포함된 각 이동체  $O_n$ 에 대해서 마지막으로 보고한 위치에 있는 세그먼트를 검색한다. 검색된 세그먼트로부터  $(QT - O_n.T_{ref})$ 시간 이내에 해당 이동 객체가 네트워크 거리로 이동했을 때 질의 영역  $QR$ 에 포함되는지를 검사하게 된다. 차량과 같은 이동체들의 이동은 도로 네트워크를 따라서 이동하기 때문에 네트

워크 세그먼트의 인접 리스트를 이용하여 이동 객체의 이동 가능한 모든 방향을 고려한다. 따라서, 보고 간격 사이에 이동 객체의 이동 방향이 바뀌더라도 이를 반영할 수 있다. 그리고 각 세그먼트를 이동하는데 소요되는 시간  $SP$ 는 해당 세그먼트 클래스의 속도로 이동했을 때 소요되는 시간으로 계산한다. 세그먼트를 이동하고 남은 시간  $RT$ 가 0이하가 될 때까지 계속해서 인접한 세그먼트를 방문하여  $QR$ 에 도달가능한지를 검사한다.

불확실 위치데이터에 대한 영역 질의의 예는 그림 14에서와 같이 루트에서부터 질의 영역  $QR$ 과 교차하는지를 검사를 함으로써 수행된다. 그림 14의 (a)는 이동체들의 마지막 보고 지점에 대한 공간 정보를 색인한 경계 사각형의 모습이며, 그림 14의 (b)는 불확실성을 반영하여 마지막 보고 시간 이후에 주어진 질의 시간까지의 경계 사각형을 확장시킨 모습이다.

그림 14의 (c)는 루트의 확장된 경계 사각형이 질의 영역과 교차하므로 이에 대한 하위 노드에 대해서도 경계 사각형을 확장시켜서 질의 영역과의 교차 여부를 확인하는 모습을 보여주고 있다. 그림 14의 (d)는 단말 노드의 엔트리인 각 이동체들에 대한 질의 처리를 보여준

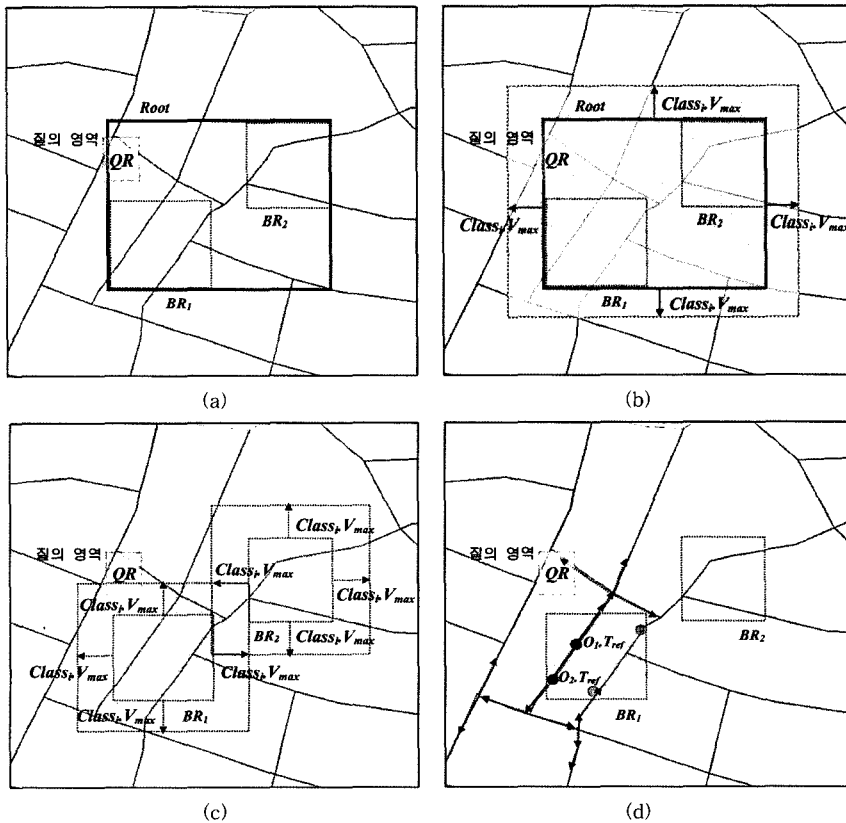


그림 14 불확실성을 반영한 영역 질의

다. 단말 노드의 각 엔트리에는 이동체들의 마지막 보고 시점의 위치 좌표와 보고 시간이 있으므로 이동체가 질의 시간  $QT$ 까지 이동 가능한 영역을 검색할 수 있다. 예를 들어, 이동체  $O_i$ 의 마지막 보고 시간 이후 이동한 시간  $RT = (QT - O_i.T_{ref})$ 이다. 이동체가 해당 세그먼트를 이동하는데 소요되는 시간을  $ST$ 라 할 때  $ST$ 는 각 네트워크 세그먼트의 거리와 최대 허용 속도로써 계산할 수 있다. 이동체의 속도 변화에 대한 불확실성을 반영하기 위해서 이동체가 허용 가능한 최대 속도로 이동했을 때의 속도로 세그먼트의 이동 소요 시간을 계산한다. 따라서, 세그먼트의 이동 소요 시간  $ST$ 를  $RT$ 에서 빼주면서 이웃 세그먼트를 검색하게 된다. 즉,  $RT = RT - ST$ 를 이용하여 0이하가 될 때까지 계속해서 이웃 세그먼트를 검색하고, 질의 영역에 속한 세그먼트에 도달 가능하다면 후보 객체로 반환한다. 이와 같은 네트워크를 이용한 영역 질의 처리에 대한 알고리즘은 다음

장에서 기술한다.

## 5. 알고리즘

이 장에서는 4장에서 제안된 도로 네트워크를 이용한 불확실 위치데이터의 영역 질의를 처리하기 위한 질의 처리 알고리즘을 제시한다.

이동체의 불확실 위치데이터를 처리하기 위해서 제안한 이동체 색인 구조에서는 이동체의 보고된 속도 정보를 사용하지 않고 위치 좌표와 보고 시간만을 사용하여 질의를 처리한다. 따라서, 알고리즘 1과 같이 기존 다차원 색인인 R-tree에서 데이터의 삽입이나 분할로 인하여 변경된 MBB를 상위 노드에 전파하는 AdjustTree 알고리즘에 시간 정보의 갱신을 추가한 구조를 사용하고 있다.

알고리즘 2는 색인의 분할 시에 새로운 노드를 생성하고 상위 노드의 변경 갱신 작업을 수행하는 알고리즘

---

### Algorithm AdjustTree(Node node)

---

```

1      BEGIN
2      IF node is the root THEN return
3      ELSE
4          // Find the parent of the node
5          parent = GetParent(node)
6          // Adjust the entry of the node in the parent
7          IF AdjustEntry(parent, [node.mbb, node.id, node.Tref]) THEN
8              // Adjust the path for the parent because the entry has been modified
9              Invoke AdjustPath(parent)
10         END IF
11     END IF
12     END

```

---

알고리즘 1 AdjustTree

---

### Algorithm SplitAndAdjust(Node node)

---

```

1      BEGIN
2          // Create a new node and distribute the entries
3          newnode = Split(node)
4          IF node is the root THEN
5              Invoke CreateNewRoot(node, newnode)
6          ELSE
7              // Get the parent of the node
8              parent = GetParent(node)
9              // Adjust the entry of the node in the parent
10             Invoke AdjustEntry(parent, [node.id, node.mbb, node.Tref])
11             // Insert the new node in the parent
12             Invoke InsertInNode(parent, [newnode.mbb, newnode.id, newnode.Tref])
13             IF parent overflows THEN
14                 Invoke SplitAndAdjust(parent)
15             ELSE
16                 Invoke AdjustPath(parent)
17             END IF
18         END IF
19     END

```

---

알고리즘 2 SplitAndAdjust

이다. 이 알고리즘에서도 마찬가지로 노드가 분할되었을 때 기존 노드와 새로이 생성되는 노드의 참조 시간을 갱신해주고 있다.

알고리즘 3은 제안하는 질의 처리 구조에서 네트워크 질의를 수행하기 전에 이동체 색인에서 후보 객체들을 검색하기 위한 알고리즘이다. 기존의 다차원 색인인 R-tree에서의 영역 질의 처리 방법과 비슷한 구조로 질의를 처리하고 있으며 다른 점은 일정한 시간 간격의 보고에 따른 이동체들의 불확실 위치데이터들을 반영하기 위한 방법으로 이동체들의 마지막 보고된 위치 좌표들을 포함하기 위해 네트워크 세그먼트의 최대 허용 속도 중에서 가장 큰 속도 값으로 각 방향으로 확장시킨 MBB를 사용하는 점이다. 따라서, 루트 노드에서부터 주어진 질의 영역  $QR$ 과 겹치는 확장된 모든 엔트리에 대하여 하위 노드로 내려가서 동일한 작업을 반복한다.

알고리즘 4는 제안한 질의 처리 구조에서 도로 네트워크를 이용하여 불확실 위치데이터를 가지고 있는 이동체들의 영역 질의를 처리하기 위한 알고리즘이다.

질의 처리는 우선 제안한 이동체 색인에서 Euclidean-RangeQuery 알고리즘을 사용해 주어진 질의 영역  $QR$ 과 겹치는 후보 객체들  $CS$ 를 검색한다. 그리고 네트워크 공간 정보 색인에서 주어진 질의 영역  $QR$ 과 겹치는 네트워크 세그먼트 집합  $NS$ 를 검색하고 후보 객체  $CS$ 에 포함된 각 이동체에 대하여 FindSegment 함수를 사용하여 해당 이동체가 속한 세그먼트  $netsegment$ 를 검색한다. 만약 이동체가 속한 세그먼트가 질의 영역에 속한 세그먼트 집합  $NS$ 에 포함된다면 해당 이동체를 질의 결과 집합  $RS$ 에 포함시킨다.  $NS$ 에 포함된 세그먼트가 아니라면 이동체  $MO$ 의 현재 위치에서 세그먼트의 양 끝점까지의 거리와 소요시간을  $D_{start}$ ,  $D_{end}$ ,  $ST_{start}$ ,

---

Algorithm EuclideanRangeQuery(Node  $node$ , QueryRange  $QR$ , QueryTime  $QT$ )

---

```

1      BEGIN
2      FOR each entry  $E$  of  $node$ 
3           $E.mbb = GetExpandedMbb(QT)$ 
4          IF  $E.mbb$  is intersect with  $QR$  THEN
5              Invoke  $EuclideanRangeQuery(cnode, QR, QT)$ , where  $cnode$  is the childnode of
                 $node$  pointed by  $E$ 
6          ELSE
7              Add for all entries  $E$  that intersect with  $QR$  to results
8          END IF
9      END FOR
10     END

```

---

알고리즘 3 EuclideanRangeQuery

---

Algorithm NetworkRangeQuery(Node  $node$ , QueryRange  $QR$ , QueryTime  $QT$ , ResultSet  $RS$ )

---

```

1      BEGIN
2      Candidate result set  $CS = EuclideanRangeQuery(node, QR, QT)$ 
3      Network segment set  $NS = RangeSearch(QR)$ 
4      FOR each moving object  $MO$  in  $CS$ 
5          Network segment  $netsegment = FindSegment(MO)$ 
6          IF  $NS$  contains  $netsegment$  THEN
7               $RS += MO$ 
8              continue
9          ELSE
10             Calculate network distances  $D_{start}$ ,  $D_{end}$  and spending times  $ST_{start}$ ,  $ST_{end}$  from the
                    network segment for each side
11             Calculate remained times  $RT_{start} = QT - ST_{start}$ ,  $RT_{end} = QT - ST_{end}$ 
12             IF  $RT \geq 0$  THEN
13                 IF  $TraverseNetwork(netsegment, RT_{start}, NS)$  or  $TraverseNetwork(netsegment,$ 
                             $RT_{end}, NS)$  THEN
14                      $RT += M$ 
15                 END IF
16             ENDIF
17         ENDIF
18     END FOR
19     END

```

---

알고리즘 4 NetworkRangeQuery

$ST_{end}$ 를 계산한다. 그리고 질의 시간  $QT$ 에서  $ST$ 의 차를 이동체의 이동 가능한 시간  $RT$ 로 설정하고  $RT$ 가 0 이하가 될 때까지 이웃 세그먼트들을 방문하면서  $NS$ 에 포함되는 세그먼트에 도달가능한지를 검색한다.

알고리즘 5는 세그먼트의 인접 리스트 정보를 이용하여 이웃 세그먼트들을 방문하고 해당 이웃 세그먼트까지의 소요 시간  $ST$ 와 이동 후 남은  $RT$ 시간을 이용하여  $RT$ 를 갱신하고 검색된 이웃 세그먼트가  $NS$ 에 포함되는지를 검사한다. 만약  $NS$ 에 포함이 되면 참을 반환한다.

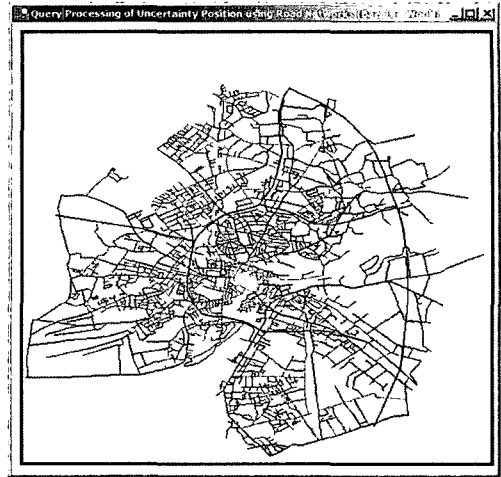
### 6. 성능 평가

이 장에서는 논문에서 제안하는 도로 네트워크를 이용한 불확실 위치데이터의 질의 처리 기법의 성능을 평가한다. 평가 대상으로 TPR-tree를 이용하여 불확실 영역을 반영하지 않은 질의 처리 방법과 [16]에서 제안한 이동체의 최대 오차만큼 질의 영역을 확장시키는 방법을 적용하였을 때 정확도를 비교한다. 실험에 사용한 도로 네트워크 데이터는 독일의 oldenberg시의 데이터를 사용하였고, 이동체의 위치데이터는 brinkhoff의 네트워크 데이터 생성기[17]를 사용하였다.

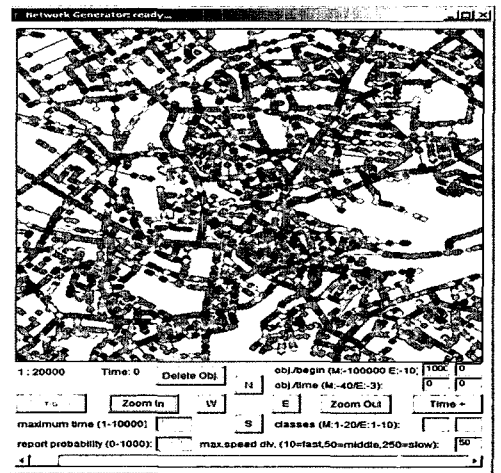
#### 6.1 도로 네트워크 및 이동체 데이터 생성

이 논문에서 사용한 네트워크 데이터는 oldenberg시의 네트워크 데이터의 node와 edge정보를 사용하여 그림 13과 같은 네트워크 세그먼트로 생성한 것이다. 그림 15의 (a)는 실험에 사용한 oldenberg시의 도로 네트워크 데이터이며 총 네트워크 세그먼트 수는 7,035개이다.

실험에 사용한 이동체 데이터는 brinkhoff의 네트워크 데이터 생성기로 oldenberg시의 도로 네트워크를 따라 이동하는 10만개의 이동체를 생성하여 실험하였다. 그림 15의 (b)는 brinkhoff 네트워크 데이터 생성기로 이동체를 생성한 모습이며 생성 환경은 표 2와 같다.



(a) 독일 oldenberg의 도로 네트워크



(b) Brinkhoff 네트워크 데이터

그림 15 도로 네트워크 및 이동체 데이터 생성 예

Algorithm TraverseNetwork(NetworkSegment *netsegment*, RemindedTime *RT*, NetworkSet *NS*)

```

1      BEGIN
2      Adjacency list AL = GetAdjacencyList(netsegment)
3      FOR each network segment netsegment in AL
4          IF NS contains netsegment THEN
5              return TRUE
6          ELSE
7              Calculate the distance D and the spending time ST
8               $RT = RT - ST$ 
9              IF  $RT \geq 0$  THEN
10                 TraverseNetwork(netsegment, RT, NS)
11             END IF
12         END IF
13     END FOR
14     END
    
```

표 2 생성 파라미터

초기 이동체 수	보고 횟수	이동 속도	비고
1,000	100	middle	주기당 생성 이동체 0

### 6.2 질의 데이터 생성

이 논문에서는 이동체의 현재 및 미래 위치에 대한 영역 질의에 대한 정확성을 평가하였다. 질의 크기는 oldenberg시의 공간 영역에 대해서 1%, 5%, 10%, 20%의 크기를 갖는 질의 1,000개를 생성하여 영역 질의의 정확도를 측정하였다. 그림 16은 전체 영역에 대하여 1%, 5%, 10%, 20%의 영역을 가지는 랜덤하게 생성한 1,000개의 질의 데이터를 나타낸 것이다.

### 6.3 실험 평가

표 1에서 *False hit*와 *False miss*에 대해서 아래와 같은 수식으로 정의를 하였다.

$$False\ hit(\%) = \frac{P - H}{P} \times 100(\%)$$

$$False\ miss(\%) = \frac{R - H}{R} \times 100(\%)$$

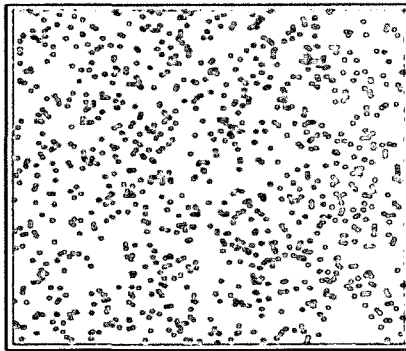
위의 수식에 따라서, *False hit*는 질의 처리 후 반환된 결과 집합에 실제 질의 조건을 만족하는 이동체 수

의 비율을 나타낸 것으로서 *False hit*가 10%라면 100개를 검색하였을 때 실제 질의 조건을 만족하는 이동체의 수가 90개이고 잘못 찾은 이동체의 수가 10개임을 의미한다. 또한, *False miss*는 실제 질의 조건을 만족하는 총 이동체 수와 검색 결과 집합에 실제 질의 조건을 만족하는 이동체 수와의 비율을 나타내는 것으로서 *False miss*가 10%라면 100개의 이동체 중에서 10개를 검색하지 못한 것을 의미한다.

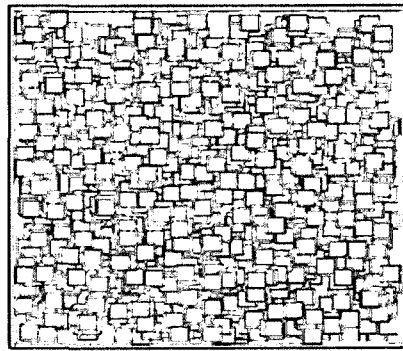
이 절에서는 불확실 영역을 반영하지 않은 TPR-tree를 이용한 질의 처리 기법(TPR\_QUERY), [16]에서 제안한 이동체의 최대 오차만큼 질의 영역을 확장시키는 기법(UTPR\_QUERY), 그리고 이 논문에서 제안한 질의 처리 기법(RD\_QUERY)에 대해서 에러율(*False miss*) 및 정확도(*False hit*)의 비교 평가를 수행한다.

#### 6.3.1 에러율(*False miss*) 평가

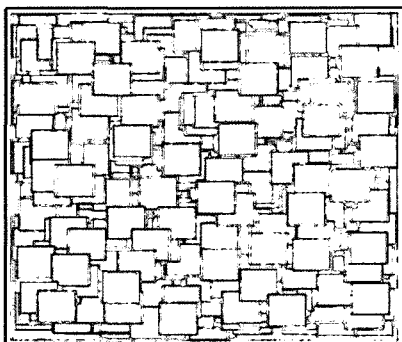
*False miss*에 대한 실험 결과 그림 17과 같이 이 논문에서 제안하는 RD\_QUERY 및 [16]에서 제안한 UTPR\_QUERY 기법은 네트워크 세그먼트 종류별로 설정된 최대 허용 속도를 넘지 않는 경우 *False miss*가 발생하지 않는 것을 확인하였다. 그리고 질의 영역을 최대 오차 크기만큼 확장하였을 때도 마찬가지로 *False miss*가



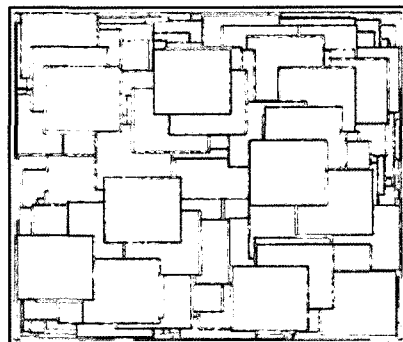
(a) 1% 질의셋



(b) 5% 질의셋



(c) 10% 질의셋



(d) 20% 질의셋

그림 16 질의 데이터 셋 생성 예

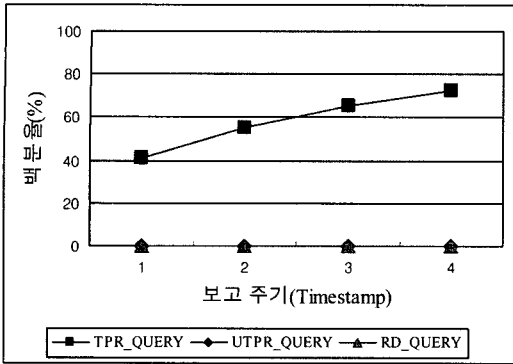


그림 17 에러율(False miss) 비교 결과

발생하지 않는 것을 확인하였다. 그러나, TPR\_QUERY의 경우 주기적인 위치 보고 시 이동체의 이동 속도 및 방향에 대한 불확실 영역이 반영되지 않으므로 보고 주기 간격이 길어질수록 False miss가 최대 70%이상 크게 증가함을 확인할 수 있다.

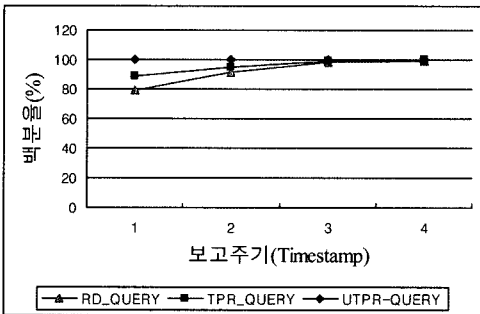
6.3.2 정확도(False hit) 평가

그림 18은 전체 영역에 대하여 1%, 5%, 10%, 20%의 질의 영역을 설정하여 질의를 하였을 때 각 질의 기법에 대한 이동체의 위치 정확도 예측 결과를 보여주고 있다. 실험의 결과 비교 대상이 되는 모든 질의 처리 기

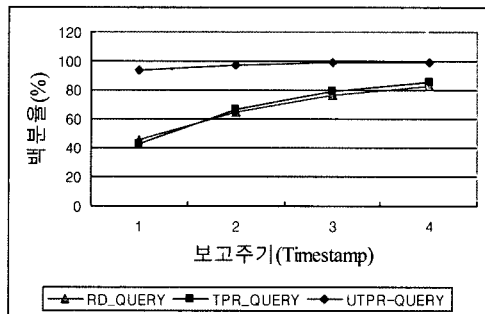
법에서 보고 주기가 짧을수록 그리고 질의 영역이 클수록 False hit가 낮게 나왔다. 이는 보고 주기가 길수록 경계 사각형의 확장된 영역 또한 커지고 검색되는 후보 객체 수가 크게 증가하기 때문이다. 그리고 질의 영역이 클수록 실제 질의 조건을 만족하는 이동체들이 질의 영역 안에 포함될 가능성이 크기 때문에 좋은 성능을 나타낸다.

이 논문에서 제안한 RD\_QUERY 기법의 False hit가 UTPR\_QUERY 기법과 비교했을 때 영역 질의와 보고 주기에 따라 최대 60% 이상 차이가 남을 알 수 있다. RD\_QUERY 기법이 TPR\_QUERY와 비슷한 수치가 나왔으나 이는 TPR\_QUERY가 상대적으로 적은 수의 후보 객체들을 검색하기 때문이다. 그러나, TPR\_QUERY의 경우 적은 수의 이동체들을 검색하기 때문에 그림 17의 실험 결과에서와 같이 False miss가 매우 높게 나옴을 확인할 수 있다.

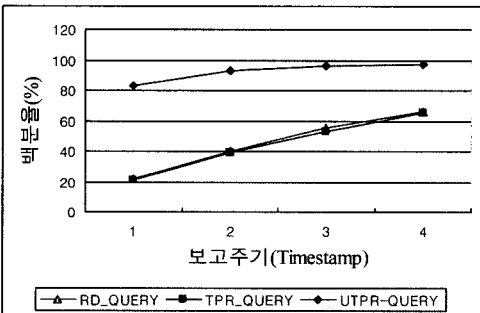
앞의 두 실험을 결과를 통해서 이 논문에서 제안한 도로 네트워크 이용한 불확실 위치데이터의 질의 처리 기법을 사용하였을 때 False hit를 낮게 유지하면서 False miss를 발생시키지 않음을 확인할 수 있다. 이는 후보 객체를 선정함에 있어 도로 네트워크의 공간 정보를 이용한 이동체들의 이동 가능한 방향과 거리를 예측할 수 있기 때문에 False hit를 최소화하였고, 네트워크



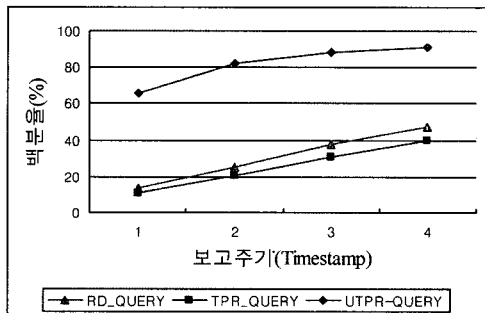
(a) 1% 영역질의



(b) 5%영역질의



(c) 10%영역질의



(d) 20%영역질의

그림 18 영역 질의에 대한 정확도(False hit) 비교 결과

세그먼트 종류별로 최대 허용 속도를 적용하고 보고 주기 사이에 이동 속도의 변화 가능한 최대 속도를 적용하여 질의를 처리함으로써 *False miss*를 피하기 때문이다.

## 7. 결론 및 향후 연구

이 논문에서는 비주기적인 위치 보고 환경에서 보고 주기 사이에 이동체의 속도와 방향의 변화로 인하여 미래 위치 질의 시에 TPR-tree와 같이 시간에 대한 선형 함수를 적용하였을 때 *False miss*와 *False hit*가 증가하는 문제점을 정의하고, 이 문제를 해결하기 위하여 도로 네트워크를 이용한 불확실 위치데이터에 대한 질의 처리 기법을 제안하였다.

보고 주기 사이에 이동체의 이동 방향의 변화에 대한 불확실성을 반영하기 위하여 네트워크 세그먼트의 공간 정보와 인접 리스트를 사용하여 정확도를 향상시키는 방법을 제안하였고, 이동체의 이동 속도의 변화에 대한 불확실성을 반영하기 위하여 네트워크 세그먼트 종류별로 이동체의 최대 허용 속도를 설정하여 정확도를 향상시키는 방법을 제안하였다. 이 논문에서 제안한 도로 네트워크를 이용한 불확실 위치데이터에 대한 질의 처리 기법은 이동체의 미래 위치에 대한 영역 질의 시에 *False miss*를 발생시키지 않으면서 *False hit*를 최소화시켰음을 실험을 통하여 확인하였다.

향후 연구로써 교통 네트워크 데이터를 사용하여 질의 정확도 및 질의 처리 속도를 향상에 대한 연구가 필요하다. 또한, 이 논문에서는 질의 처리를 하기 위한 자료 구조로써 R-tree를 사용하고 있으나 노드 중첩의 최소화를 통하여 이동체의 위치 검색 및 도로 네트워크 검색을 빠르게 하기 위한 색인 기법의 적용이 필요하다.

## 참고 문헌

- [1] O. Wolfson, A. P. Sistla, B. Xu, J. Zhou, and S. Chamberlain, "DOMINO: Databases for Moving Objects Tracking," Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, pp.547-549, 1999.
- [2] M. Erwig, R. H. Gutting, M. Schneider, and M. Vazirgiannis, "Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases," GeoInformatica, vol. 3(3), pp.269-296, 1999.
- [3] R. H. Gutting, M. H. Bohlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, and M. Vazirgiannis, "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, vol. 25(1), pp.1-42, 2000.
- [4] D. Pfoser, Y. Theodoridis, and C. S. Jensen, "Indexing Trajectories in Query Processing for Moving Objects," Chorochronos Technical Report, CH-99-3, October, 1999.
- [4] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel Approaches in Query Processing for Moving Objects," Proc. of Int'l Conf. on Very Large Data Bases, pp.395-406, 2000.
- [5] N. Beckmann and H. P. Kriegel, "The R\*-tree: An Efficient and Robust Access Method for Points and Rectangles," Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, pp. 332-331, 1990.
- [6] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez, "Indexing the Positions of Continuously Moving Objects," Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, pp.331-342, 2000.
- [7] 진희규, 김동현, 임덕성, 홍봉희, "주기적인 위치 보고 이동체의 불확실 위치 처리", 한국정보과학회 2004 가을 학술발표논문집(II), 제31권, 제2호, pp.241-243, 2004.
- [8] 배태욱, 안경환, 홍봉희, "도로 네트워크를 이용한 불확실 위치 데이터의 질의 처리", 한국정보과학회 2004 가을 학술발표논문집(II), 제31권, 제2호, pp.88-90, 2004.
- [9] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. of the ACM SIGMOD Int'l Conf. on the Management of Data, pp.47-57, 1984.
- [10] Y. Tao, D. Papadias, and J. Sun, "The TPR\*-tree: An Optimized Spatiotemporal Access Method for Predictive Queries," Proc. of Int'l Conf. on Very Large Data Bases, pp.790-801, 2003.
- [11] S. Saltenis and C. S. Jensen, "Indexing of Moving Objects for Location-Based Services," Proc. of Int'l Conf. on Data Engineering, pp. 463-472, 2002.
- [12] C. M. Procopiuc, P. K. Agarwal, and S. Har-Peled, "STAR-Tree: An Efficient Self-Adjusting Index for Moving Objects," Proc. of ALENEX, pp.178-193, 2002.
- [13] D. Pfoser and C. S. Jensen, "Capturing the Uncertainty of Moving-Object Representations," Proc. of Int'l Symp. on Advances in Spatial Databases, pp.111-132, 1999.
- [14] D. Pfoser and N. Tryfona, "Capturing Fuzziness and Uncertainty of Spatiotemporal Objects," Proc. of East European Conf. on Advances in Databases and Information Systems, pp. 112-126, 2001.
- [15] J. H. Hosbond, S. Saltenis, and R. Ortoft, "Indexing Uncertainty of Continuously Moving Objects," Int'l Workshop on Databases and Expert Systems Applications, pp.911-915, 2003.
- [16] T. Brinkhoff, "A Framework for Generating Network-Based Moving Objects," GeoInformatica, vol.6(2), pp.153-180, 2002.



안 성·우

1999년 부산대학교 컴퓨터공학과 졸업(공학사). 2001년 부산대학교 대학원 컴퓨터공학과 졸업(공학석사). 2001년~현재 부산대학교 대학원 컴퓨터학과 박사과정. 관심분야는 지리정보 시스템, RFID 미들웨어, 모바일 GIS, 이동체 색인



안 경 환

1997년 부산대학교 컴퓨터공학과 졸업(공학사). 1999년 부산대학교 대학원 컴퓨터공학과 졸업(공학석사). 2004년 부산대학교 대학원 컴퓨터공학과 졸업(공학박사). 2004년~현재 한국전자통신연구원 LBS연구팀 연구원. 관심분야는 LBS, 이동체 데이터베이스, 스트림데이터처리



배 태·욱

2003년 한국해양대학교 컴퓨터공학과 졸업(공학사). 2005년 부산대학교 대학원 컴퓨터공학과 졸업(공학석사). 2005년~현재 LG전자 단말연구소 주임연구원. 관심분야는 LBS, 유비쿼터스 센서 네트워크

홍 봉·희

정보과학회논문지 : 데이터베이스  
제 33 권 제 1 호 참조