

시공간 질의 처리를 위한 태그 객체의 모델링 기법

김동현*, 반재훈**

*동서대학교 소프트웨어전문대학원, **경남정보대학 인터넷응용계열
pusrover@dongseo.ac.kr, chban@kit.ac.kr

The Modeling Scheme of Tag Objects for processing Spatio-Temporal Queries

DongHyun Kim, ChaeHoon Ban

Graduate School of Software, DongSeo University

Subdivision of Internet Application, KyungNam College of Information & Technology⁺

요 약

RFID는 라디오 주파수 기반의 자동 인식 데이터 수집 장치로서 유비쿼터스 컴퓨팅 기반 기술의 핵심 분야이다. RFID 태그(전자태그)가 부착된 개체의 위치를 저장할 수 있기 때문에 시간에 따라 위치가 변하는 이동체와 유사하다. 그러나 태그 객체는 판독기(reader)의 위치로 인식되며 일정 수 이상의 객체들이 같은 시간에 함께 이동하는 특징을 가지기 때문에 기존의 이동체 모델링 방법을 사용하여 전자 태그 객체를 표현하면 색인의 성능이 저하되는 문제가 발생한다.

이 논문에서는 시공간 질의를 처리하는 색인의 성능을 개선하기 위하여 태그 객체의 특징을 반영한 모델링 기법을 제안한다. 제안한 모델링 기법에서 간격 데이터 개념을 기반으로 시간과 공간 도메인 외에 식별자 도메인을 추가하여 태그 객체를 표현한다. 그리고 세가지 도메인을 이용하여 태그 객체에 대한 색인을 구성한다.

1. 서론

RFID(Radio Frequency Identification)는 라디오 주파수를 이용하여 실 개체의 데이터를 수집하기 위한 자동 인식 데이터 수집 장치로서 유비쿼터스 컴퓨팅 환경을 구축하기 위한 핵심기술이다. RFID 시스템은 물류, 재고 관리 분야에서 컨테이너, 팔레트와 같은 실 개체에 부착되어 개체의 위치를 RFID 리더를 통하여 기록할 수 있다. 따라서 RFID가 장착된 개체인 태그 객체의 과거 이동 과정뿐만 아니라 현재 위치도 추적 관리할 수 있는 위치 기반 서비스가 가능하다[8][9]. 그러나, 이러한 서비스를 위해서는 예를 들어 “2004년 10월 29일부터 10월 30일까지 정문 #4를 통과한 컨테이너를 검색하라”와 같은 시공간 질의를 지원해야 하기 때문에 시공간 질의 처리를 위하여 태그 객체에 대한 연구가 필요하다.

태그 객체는 이동체와 유사하게 시간에 따라 객체의 위치 정보가 변하며 그 외에 다음의 고유 특성을 가진다. 첫째, 이동체의 위치는 리더기의 위치에 의해 정해진다. 둘째, 다수의 객체가 동일한 이동 궤적을 가진다. 셋째, 리더기간의 간섭에 의하여 동일한 객체에 대하여 각기 다른 위치에 대하여 시간의 중복이 발생한다. 마지막으로 리더기가 성긴(coarse) 경우 태그 객체의 이동 궤적에 단절 현상이 발생한다.

따라서 이러한 특성을 가진 태그 객체를 기존의 이동체 모델링 기법을 이용하여 표현하여 색인할 경우 다음의 문제가

발생한다. 태그 객체의 위치는 리더기의 위치에 의해 결정되기 때문에 리더기의 인식 범위내에 다수의 태그 객체가 위치할 경우 동일 범위 내의 모든 객체는 동일한 위치 데이터를 가지게 된다. 또한 대부분의 태그 객체가 인접한 시간동안 리더기의 인식 범위내에 존재한다. 따라서 태그 객체들이 이동체 모델링 기법 및 색인 기법을 이용하여 색인할 경우 노드간의 겹침이 심하여 색인의 성능이 저하되는 문제가 있다.

시공간 질의를 처리하기 위한 태그 객체의 모델링 및 색인 기법의 기존 연구는 없다. 그러나 유사한 특성을 가지고 있는 이동체를 색인하기 위한 연구로는 R-Tree를 기반으로 한 3DR-Tree[3], 2+3 R-Tree[6], HR-Tree[7] 등이 있다. 3DR-tree에서는 공간외에 시간 도메인을 추가하여 객체를 색인한다. 2+3 R-tree는 2차원 점 객체와 3차원 선분 객체를 따로 저장하고 두 개의 색인 구조를 사용하여 색인한다. 그리고 HR-tree는 객체의 이력 정보를 유지하여 각 타임스탬프마다 색인 구조를 유지한다. 그러나 기존의 색인 구조는 태그 객체의 특성을 고려하지 않았기 때문에 노드간의 겹침이 심해지는 문제가 있다.

이 논문에서는 시공간 질의를 처리하는 색인의 성능을 개선하기 위하여 전자태그 객체의 특징을 반영한 모델링 및 색인 기법을 제안한다. 제안한 모델링 기법에서는 전자태그 객체를 간격 데이터로 표현하고 시간과 공간 도

메인 외에 식별자 도메인을 추가하여 태그 객체의 색인을 구성한다. 이를 위하여 태그 객체의 특성과 질의를 분석한다. 그리고 R-tree를 기반으로 태그 객체를 색인하기 위한 색인 구조를 제시한다.

2장에서는 이동체 색인 구조에 관한 관련연구를 기술하고 3장에서 기존 이동체 모델링 및 색인 기법의 문제점을 정의한다. 4장에서 태그 객체의 특성 및 객체에 대한 대상 질의를 분석하고 5장에서 특성을 고려한 태그 객체의 모델링 기법을 제시한다. 그리고 6장에서 R-tree 기반의 색인 구조를 제시하고 마지막으로 7장에서 결론을 기술한다.

2. 관련연구

3차원 공간 모델에서 이동체에 관한 색인 구조로는 3D R-Tree[3], 2+3 R-Tree[6], HR-Tree[7] 등이 있다. 기존 R-Tree에 기반한 3DR-Tree는 1차원 시간과 2차원 공간을 따로 분리하여 저장하지 않고 하나의 3차원 색인에 저장한다. 그러나 기존 R-tree의 삽입 및 분할 정책을 사용하기 때문에 시간에 따라 위치 데이터가 변하는 연대기적 데이터를 색인하면 비효율적인 공간활용도를 보인다. 또한 쿼리 선택의 두 끝점 중 한 점의 데이터를 알 수 없을 때 선택을 색인할 수 없는 문제가 있다.

2+3 R-Tree는 3DR-Tree의 문제점을 2차원 점과 3차원 선분에 대해 각각 따로 저장하는 두 개의 다른 R-Tree를 사용함으로써 해결한다. 2차원 점은 현재 시점에서 객체의 공간상의 점을 표현하며, 3차원 선분은 객체의 과거 이력을 표현한다. 그러나, 2+3 R-Tree는 두 개의 R-Tree에서 질의를 수행해야 하는 문제점이 있다.

HR-Tree는 트랜잭션 시간 개념을 추가하여 객체의 이력 정보를 표현하고 각 타임스탬프마다 색인 구조를 유지한다. 이 때 상태가 변한 루트와 가지를 대체함으로써 현재와 과거를 유지하며, 상태가 변하지 않는 가지들은 복제하지 않는다. 그러나 기존의 이동체를 위한 모델링 및 색인 구조들은 태그 객체의 특성을 고려하지 않았기 때문에 태그 객체들이 인접한 시간대에 몰려있는 경우 노드간의 겹침이 심해지는 문제가 있다.

3. 문제 정의

기본적인 전자 태그 시스템에서 각각의 리더기는 일정 영역내의 전자 태그를 인식할 수 있는 호출영역(Interrogation zone)을 가진다[8][9]. 전자 태그 객체 to_1 이 리더기 R_1 의 호출영역내에 들어가면 Enter 이벤트가 발생한다. Enter 이벤트가 발생하면 R_1 은 to_1 에게 에너지를 전송하고 to_1 은 전송받은 에너지를 이용하여 자신의 인식자(tid)를 리더기에게 전송한다.

따라서 to_1 이 R_1 의 호출영역내에 위치하고 있는 동안에는 R_1 은 to_1 의 인식자를 전송받아 인식자를 서버에게 전달할 수 있다. 만약 to_1 이 R_1 의 호출영역을 벗어나면 Leave 이벤트가 발생하고 R_1 은 to_1 의 인식자를 더 이상 전송받을 수 없다.

리더기는 특성상 자신의 호출영역내에 위치하고 있는 태그 객체의 정확한 위치 정보를 정할 수 없고 자신의 호출영역내에 특정 태그 객체의 존재 여부만을 파악할 수 있다. 따라서 태그 객체의 정확한 x, y 그리고 z 위치 데이터를 획득할 수 없기 때문에 객체의 위치는 리더기의 위치 좌표에 의해 결정된다. 또한 이동체와는 달리 태그 객체는 응용에 따라 리더기들의 호출영역에서 이동할 경우 다수의 객체가 동시에 이동한다.

만약 다수의 태그 객체가 인접한 시간대에서 이동하면 해당 집합의 태그 객체들은 동일한 위치 좌표와 인접한 시간값을 가지게 된다. 이러한 태그 객체를 트리 구조의 색인을 사용하여 위치 도메인과 시간 도메인에 따라 색인할 경우에 노드간의 겹침이 매우 심해진다. 따라서 태그 객체의 삽입 또는 검색 연산 시 색인의 성능이 저하되는 문제가 발생한다. 예를 들어 최대 엔트리 수가 4인 R-Tree 색인 구조에서 판독기 R_1 에 위치한 16개의 태그 객체를 삽입하면 그림 1과 같은 구조가 나온다. 그러나 비단말 노드는 모두 동일한 위치 정보를 가지고 있기 때문에, R_1 을 포함하는 영역에 대한 질의를 처리하기 위해서는 그림 1의 전체 트리를 검색해야 하므로 최소 11번의 I/O가 필요하다.

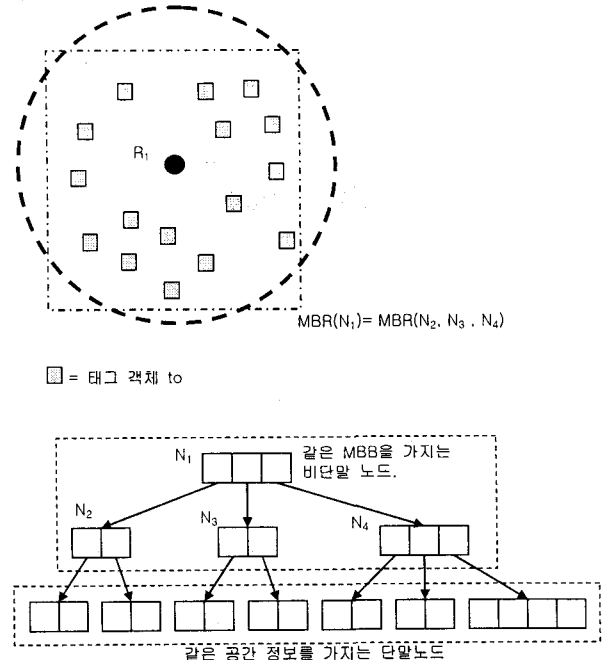


그림 1 태그 객체 색인의 예

4. 태그 객체 특성

4.1 태그 객체의 특성

전자 태그를 부착한 실 개체는 시간이 지남에 따라 위치가 바뀌기 때문에 시간에 따라 이동하는 이동체와 유사한 특성을 가진다. 그러나 단위 시간당 움직임이 빈번한 이동체와는 달리 시간당 이동이 적고 위치 데이터가 부정확한 태그 객체는 다음의 고유 특성을 가지게 된다. 첫째로 태그 객체의 위치는 리더기의 위치에 의해 결정되기 때문에 태그 객체는 논리적 위치 데이터를 가진다. 특성상 리더기는 자신의 호출 영역내에 위치한 태그의 인식만 가능하고 영역내의 정확한 위치는 알 수 없다. 따라서 그림 2에서 보듯이 t_{01} 이 리더기 R_1 에서 R_2 로 이동할 때 t_{01} 의 위치는 리더기들의 위치 데이터로 표현된다.

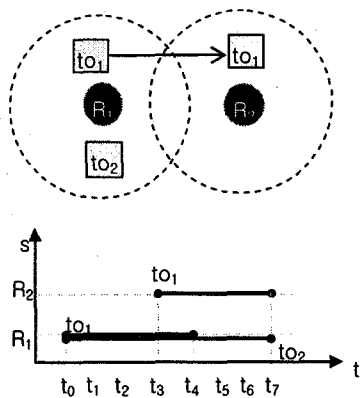


그림 2 태그 객체 t_0 의 논리적 위치 표현

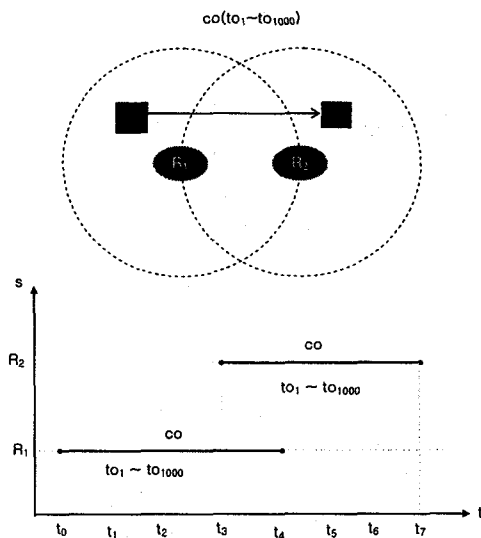


그림 3 태그 객체 $t_{00} \sim t_{1000}$ 의 동일 궤적

둘째로 태그 객체는 다수의 객체가 동일한 이동 궤적을 가진다. 물류 분야에서 전자 태그가 부착된 일반 상품의 경우에 팔레트 또는 컨테이너등의 보관함에 적재되어 한꺼번에 이동된다[8][9]. 따라서 하나의 집합화된 태그 객체들은 리더

기의 호출 영역내에서 동시에 이동하기 때문에 그림 3에서 보듯이 동일 집합에 소속된 모든 태그 객체들은 위치 데이터의 동일한 이동 궤적을 가지게 된다.

세 번째 특성은 만약 리더기들의 위치가 서로 밀집되면 리더기간의 간섭으로 인하여 동일 태그 객체의 위치가 중복된다. 마지막으로 네 번째 특성은 만약 리더기들의 위치가 성기면 태그 객체의 이동 궤적에 단절이 발생하게 된다.

4.2 태그 객체 대상 질의

태그 객체를 대상으로 하는 질의는 크게 4가지 종류로 구분될 수 있다. 첫 번째는 특정 태그의 인식자와 시간을 이용하여 태그가 위치한 위치 데이터를 검색하는 find 질의이다. 예를 들어 태그 객체 #13이 2004년 10월 29일에서 10월 30일 사이에 위치한 곳의 위치(리더기의 위치)를 검색하는 질의는 find 질의에 해당한다. 두 번째는 태그 객체의 인식자와 시간을 이용하여 같은 위치에 위치하였던 다른 태그 객체들을 검색하는 with 질의이다. 세 번째는 리더기의 위치와 시간을 이용하여 특정 리더기에 위치한 태그들을 검색하는 look 질의이다. 예를 들어 리더 #16에 2004년 9월 20일에서 24일 사이에 위치하였던 태그들을 검색하는 질의는 look 질의이다. 마지막으로 태그 객체의 인식자를 이용하여 특정 태그의 이동 경로를 검색하는 history 질의이다. 각각의 질의에 대한 분류는 표 1과 같다.

표 1 태그 대상 질의의 분류

질의 명	input	output
find	tag id, time	location
with	tag id, time	a set of tag ids
look	location, time	a set of tag ids
history	tag id	a set of locations

5 태그 객체 모델링

5.1 태그 객체 모델링 기법의 분류

4.2절에서 분류한 대상 질의 종류에 따라 태그 객체를 표현하기 위한 기법은 크게 세가지로 구분될 수 있다. 첫 번째는 find 질의를 처리하기 위한 태그 중심 모델(tag centered model)이다. 이 모델에서 태그 객체는 태그 인식자와 시간을 이용하여 표현되며 시간 도메인과 태그 인식자 도메인을 이용하여 색인된다. 특히 태그 인식자 도메인의 해상도(resolution)는 색인되는 태그 객체의 수에 따라 결정되기 때문에 특정 객체는 특정 시간에 하나 또는 두 개 이하가 되는

장점이 있다. 태그 중심 모델은 태그 인식자를 하나의 도메인으로 설정하고 있기 때문에 find 질의와 with 질의는 효과적으로 처리할 수 있으며 history 질의도 처리 가능하다. 그러나 look 질의는 태그 객체의 수만큼 검색해야 하기 때문에 전체 색인을 검색해야 하는 단점이 있다.

두 번째는 look 질의를 처리하기 위한 위치 중심 모델(location centered model)이다. 이 모델에서 태그 객체는 시간과 리더기의 위치 데이터를 이용하여 표현되며 시간 도메인과 위치 도메인을 이용하여 색인된다. 특히 리더기의 위치 데이터인 위치 도메인의 해상도는 리더기의 개수에 의하여 결정되기 때문에 특정 위치의 특정 시간대에 여러 개의 동일한 데이터가 존재하는 단점이 있다. 이 모델은 리더기의 위치와 시간 도메인을 이용하여 색인하기 때문에 look 질의를 효과적으로 처리하는 장점이 있다. 또한 find 질의와 with 질의는 시간을 이용하여 검색한 후에 태그 인식자를 이용하면 처리 가능하다. 그러나 특정 태그 인식자만을 이용하는 history 질의를 처리하기 위해서는 전체 색인을 검색해야 하는 단점이 있다.

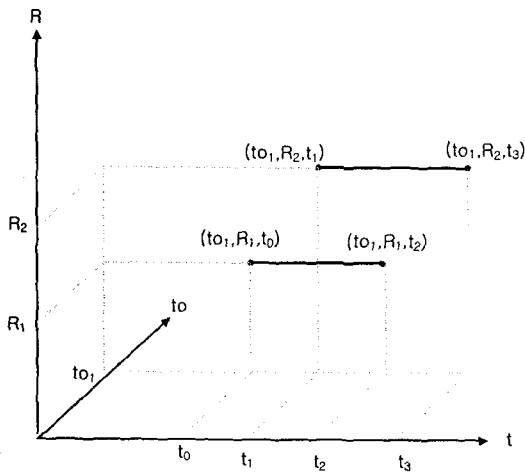


그림 4 태그/위치 중심 모델의 색인 예

마지막으로 세 번째는 앞서 기술한 두 가지 모델을 병합한 태그/위치 중심 모델(tag/location centered model)이다. 태그 객체는 시간, 리더기의 위치 그리고 태그 인식자를 이용하여 표현되며 시간과 위치, 태그 인식자인 세 가지 도메인을 이용하여 그림 4와 같이 색인된다. 이전에 기술된 두 가지 모델의 병합 모델이므로 위치 도메인은 설치된 리더기의 개수에 의하여 해상도가 결정되며 태그 인식자 도메인은 태그 객체의 수에 의하여 결정된다. 태그/위치 중심 모델은 리더기의 위치뿐만 아니라 태그 인식자와 시간 도메인에 대해서도 색인하기 때문에 find, with, look 그리고 history 질의를 모두 처리할 수 있는 장점을 가진다. 따라서 이 논문에서는 태그/위치 중심 모델을 이용하여 태그 객체를 표현한다.

5.2 간격 데이터를 이용한 태그 객체 표현

각 객체는 도메인이나 데이터 타입 또는 값을 가지는 속성을 가지기 때문에 먼저 이를 위한 캐리어 집합(carrier set) A_O (데이터 타입 O 의 캐리어 집합)를 정의해야 한다. 3차원 순수 공간 객체 타입으로는 점 point와 시간 도메인에 속하는 time이 있다. 그리고 판독기 객체 reader는 미리 정해진 3차원 상의 점에 위치한다. 각각의 도메인에 대한 캐리어 집합의 정리는 다음과 같다.

정리 1: 각 도메인에 대한 캐리어 집합

- $A_{point} = \text{real} * \text{real} * \text{real}$
- $A_{time} = \text{real}$
- $A_{reader} = \{(x_i, y_i, z_i) | (x_i, y_i, z_i) \in A_{point}\}$

이 논문에서는 태그 객체와 관련된 기본적인 질의인 find와 look을 처리하기 위해서 태그/위치 중심 모델의 개념을 이용하여 태그 객체를 간격(interval) 데이터로 정의한다. 이러한 데이터는 객체 식별자(tid), 판독기의 위치(loc), 시간 간격(time interval)으로 구성되고 하나의 다차원 색인에 저장된다. 이때 find 질의를 효율적으로 처리하기 위해서 위치와 시간 뿐만 아니라 객체 식별자도 색인 도메인에 포함시킨다. 따라서 태그 객체의 정의는 다음과 같다.

정리 2: 태그 객체 정의

- $A_{tag} = \{(tid_j, loc_j, t_{enter}, t_{leave}) | t_{enter}, t_{leave} \in A_{time}, t_{enter} < t_{leave}, loc_j \in A_{reader}\}$

색인에 저장되는 데이터는 $(tid, loc, t_{enter}, t_{leave})$ 로 표현되며 시간 간격 $[t_{enter}, t_{leave}]$ 동안 객체 tid가 판독기의 위치 loc에 존재했음을 의미한다. tid가 판독기의 위치 loc에서 enter를 발생시키면 새로운 데이터가 색인에 삽입된다. 이때 인식 영역을 벗어나는 시간(t_{leave})을 알 수 없기 때문에 시간 간격에서 t_{leave} 는 now로 나타낼 수 있으며, 현재까지 인식 영역 안에 존재함을 의미한다. 반면 객체 tid가 loc에서 leave 이벤트를 발생시킬 때는 기존 데이터에서 now의 값을 t_{leave} 로 갱신하는 연산이 수행된다. 구현에서 now의 값은 시간 도메인에서 가장 큰 값을 사용한다.

now 값을 가지는 현재 위치 데이터는 갱신 연산이 수행될 때까지 시간이 지남에 따라 계속 성장한다(갱신 연산이 수행된 데이터는 더 이상 성장하지 않는다). 이 논문에서는 두 데이터를 구별하기 위해서 now 값을 포함하는 간격 데이터를 성장 간격(growing interval, GI)이라 하고 갱신 연산이 수행된 데이터를 고정 간격(fixed interval, FI)이라 한다.

6. 태그 객체의 색인

6.1 색인 구조

색인 구조의 비단말 노드는 기존 R-Tree와 동일한 $\langle cp, rect \rangle$ 형태의 엔트리를 가진다. cp 는 트리에서 자식 노드의 주소를 가리키고 $rect$ 는 자식 노드의 모든 엔트리를 포함하는 최소 경계 박스(Minimum Bounding Box, MBB)를 의미한다. 단말 노드는 $\langle \rangle$ 형태의 엔트리를 가진다. I 는 $\langle tid, loc, t_{enter}, t_{leave} \rangle$ 형태이기 때문에 tid 를 단말 노드의 엔트리에는 따로 저장할 필요가 없다. loc 는 2차원일 경우 $(x, y, 0)$ 의 공간 좌표를 가지고, 3차원일 경우 (x, y, z) 를 가진다.

색인을 변경하는 주요 연산은 $enter$ 이벤트가 발생할 때 수행되는 삽입 연산이며 이 연산은 노드의 분할을 일으키고 색인의 크기를 증가시킨다. $leave$ 이벤트가 발생할 경우, 기존 데이터를 수정하는 갱신 연산이 수행되며, 색인의 크기는 변경되지 않는다. 대신에 단말 노드의 엔트리에서 t_{leave} 값이 큰 값에서 작은 값으로 수정되기 때문에 필요한 경우 단말 노드에서 루트 노드까지 노드의 MBB를 조절한다.

6.2 데이터 삽입

삽입 알고리즘은 새로운 데이터 G_I 가 들어갈 단말 노드를 찾는 ChooseSubtree와 삽입 후의 노드 MBB를 조절하고 분할을 상위 노드로 전파하는 AdjustTree 알고리즘으로 구성된다. R-Tree에서 삽입은 최소 영역 확장을 가지는 엔트리를 선택하여 단말 노드를 찾아 내려 간다. 그러나 최소 영역 확장 정책은 간격 데이터에서 노드 겹침을 증가시킬 수 있는데, 이것은 노드가 G_I 와 F_I 를 동시에 가질 수 있기 때문이다. 그림 5은 새로운 데이터 $G_{I_{NEW}}$ 가 삽입되면서 N_1 보다 N_2 에서 더 작은 영역 확장을 가질 경우 N_1 과 N_2 에서 노드 겹침이 발생함을 보여준다. 이를 해결하기 위해서는 최소한의 노드 겹침을 일으키는 엔트리를 선택해야 한다. 최소한의 노드 겹침을 가지는 엔트리를 선택하는 알고리즘은 기존의 논문에 있기 때문에 이 논문에서는 삽입 알고리즘을 따로 기술하지 않는다.

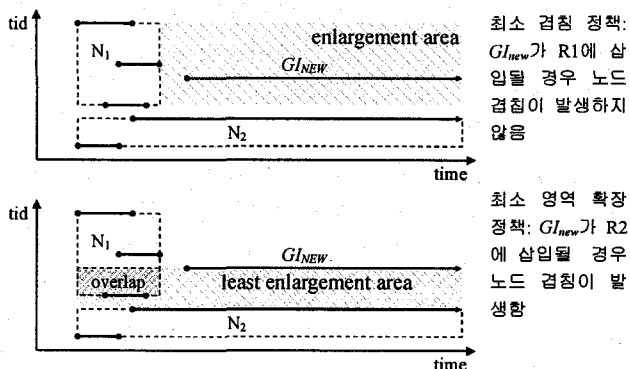


그림 5 최소 영역 확장 정책으로 인한 노드 겹침

6.3 단말 노드 분할 알고리즘

G_I 와 F_I 는 색인의 단말 노드에 동시에 존재할 수 있으므로

노드의 영역을 크게 만드는 G_I 보다는 F_I 를 많이 가짐으로써 이웃 노드와의 겹침을 최소화해야 한다. 분할 시에 단말 노드에 있는 tid 수는 G_I 의 수와 같거나 작기 때문에 tid 수가 작으면 F_I 수가 높아진다. 이 논문에서는 단말 노드에 존재하는 tid 수를 작게 만들기 위해서 임의의 p 개의 tid 를 가질 때까지 tid 축에서 분할하는 정책을 사용한다. 여기서 p/N (N : 노드의 최대 엔트리 수)를 tid 분할 인자(TSF)라 한다.

단말 노드에서 분할이 일어나는 과정은 다음과 같다. 단말 노드에서 오버플로우가 발생할 경우 노드에 존재하는 tid 수가 p 보다 크면 tid 축에서 분할이 일어난다. 다음 분할 시에 tid 수가 p 보다 크면 계속해서 tid 축에서 분할이 일어나고 그렇지 않을 경우에는 시공간 분할을 수행한다. 시공간 분할은 기존 R*-tree의 분할 정책과 동일하지만 tid 값은 고려하지 않고 시간과 공간으로 형성된 MBB만 고려한다. 시공간 분할 후에 tid 축으로 분할하지 않으면 반드시 시간 축 분할을 수행하며, 시간 축 분할은 G_I 와 F_I 를 서로 다른 노드에 저장한다.

7. 결론

이동체와 유사하게 태그 객체의 위치는 시간에 따라 변화하지만 이동체와는 달리 객체의 위치가 리더기의 위치 데이터에 의해 정해지고 다수의 객체가 동일한 이동체적을 가지는 특성이 있다. 만약 기존의 이동체 모델링 및 색인 기법을 이용하여 태그 객체를 표현하고 색인하면 노드간의 겹침이 심해지는 문제가 발생한다.

이 논문에서는 시공간 질의를 효과적으로 처리하기 위하여 태그 객체의 특성을 반영한 모델링 및 색인 기법을 제안하였다. 간격 데이터의 개념을 이용하여 시간과 공간 도메인 외에 식별자 도메인을 추가하고 태그 객체를 표현하였다. 그리고 세가지 도메인을 기반으로 태그 객체의 색인을 구성하였다. 이를 위해 태그 객체의 특성과 객체에 대한 질의 유형을 분석하였다. 이 논문에서 제시한 모델링 및 색인 기법은 태그 객체에 대한 질의를 효과적으로 처리할 수 있는 장점을 가진다. 향후 연구로는 기존 이동체를 위한 색인 구조와의 실험 비교가 필요하다.

참고문헌

- [1] A. Guttman, "R-trees: A dynamic index structure for spatial searching", ACM SIGMOD Conference, pp.47-54, 1984.
- [2] N. Beckmann and H. P. Kriegel, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", In Proc. ACM SIGMOD, pp.332-331, 1990
- [3] Y. Theodoridis, M. Vazirgiannis and T. Sellis, "Spatio-Temporal Indexing for Large Multimedia Applications", In

- International Conf, on Multimedia Computing and System, pp.441-448, 1996
- [4] K. Romer, T. Schoch, F. Mattern and T. Dubendorfer, "Smart identification frameworks for ubiquitous computing applications" Pervasive Computing and Communications. Proceedings of the First IEEE International Conference o, pp256-262, 2003.
- [5] Y. Theodoridis, J. R. Silva and M. A. Nascimento, "On the Generation of Spatiotemporal Datasets", SSD, Hong Kong, LNCS 1651, Springer, pp.147-164, 1999.
- [6] M. A. Nascimento, J.R.O. Silva and Y, Theodoridis, "Evaluation of Access Structures for Discretely Moving Points.", In Proc. of the Intl. Workshop on Spatiotemporal Database Management (STDBM'99), pp. 171-188. Edinburgh, UK, Sep/99.
- [7] M. A. Nascimento and J.R.O. Silva, "Towards historical R-Trees.", In Proc. of the 1998 ACM Symposium on applied Computing, pages 235-240, February 1998.
- [8] 이기형, 서영덕, 김동현, 홍봉희, "전자태그 객체의 시공간 모델과 현재 질의를 위한 색인 구조", 한국정보과학회 데이터베이스 연구회 2004 학술발표논문집, 제 20 권 제 2 호, p133-140.
- [9] 이기형, 반재훈, 김동현, 홍봉희, "RFID 태그 객체의 간격 데이터 색인", 한국정보과학회 2004 가을 학술발표논문집, 제 31 권 2 호, pp82-84