

RFID 태그 데이터의 색인을 위한 불균형 확장 정책*

김기홍, 안성우, 홍봉희
부산대학교 컴퓨터공학과
e-mail : {buglist, swan, bhhong}@pusan.ac.kr

Disproportional Enlargement Policy for Indices of RFID Tag Data

Gihong Kim, Sungwoo Ahn, Bonghee Hong
Dept. of Computer Engineering, Pusan National University

요 약

RFID 시스템에서 태그를 부착한 객체의 위치를 추적하는 응용이 중요하며 태그 위치를 추적하기 위한 질의를 효율적으로 처리하기 위해서는 태그 궤적의 저장 및 검색을 빠르게 처리 할 수 있는 색인이 필요하다. 기존의 시공간 기반 이동체 색인과는 달리 태그 위치 색인은 태그 식별자(tid), 리더 식별자(rid), 시간(time)을 도메인으로 가진다. RFID 환경에서는 태그 위치 색인의 rid 와 tid 도메인의 크기가 크고 한쪽 도메인으로 불균등한 영역 질의의 발생이 빈번하다. 따라서, 최소 면적 확장 정책에 따른 기존의 이동체 색인의 삽입 기법을 태그 위치 색인에 그대로 적용했을 경우 질의 영역과 생성된 노드 간의 겹침이 심하게 되어 색인 검색 비용이 증가하는 문제가 발생한다.

논문에서는 R*-tree 의 삽입 정책을 따르는 태그 위치 색인의 삽입 방법으로 불균형 확장 정책을 제안한다. 제안한 삽입 정책은 들레길이에 가중치를 부여하는 방법이며 이러한 정책을 사용함으로써 불균등한 질의 영역으로 인해 발생하는 질의 영역과 노드간의 중첩을 최소화하여 검색 시 노드 접근 횟수를 줄인다. 또한 실험을 통하여 기존 삽입 방법인 최소 면적 확장 정책의 삽입 비용과 검색 비용을 비교하였으며 그 결과 불균등한 질의 영역을 가지는 RFID 환경에서 삽입 및 검색 비용을 줄여 성능을 향상시켰다.

1. 서론

최근 RFID 전자 태그가 부착된 객체의 위치 추적과 관련된 응용이 중요하게 부각되고 있으며 RFID 리더(reader)로부터 수집되어 저장되는 태그 이벤트를 효율적으로 검색하기 위해서는 색인의 구축이 필요하다.

기존 이동체의 시간의 흐름에 따른 위치 정보를 색인하는 방법은 RFID 태그 객체의 이동과 유사하여 RFID 태그 객체의 위치 추적을 위한 색인에 사용되

고 있다. 이동체의 위치 정보를 색인 하는 방법을 기반으로 최근 태그 객체의 위치 특성을 고려하여 저장 및 질의 처리를 효과적으로 처리하기 위한 색인인 TPIR-tree(Time Parameterized Interval R-tree)[1]가 제안되었다. 이 색인의 경우 삽입, 분할 시 R*-Tree[2]의 방식을 기본적으로 따르고 있다.

기존의 이동체 색인에서는 공간 좌표(x, y)와 시간(time)으로 구성된 3 차원 도메인 공간 상에 모델링된 이동체의 궤적을 저장하지만 RFID 위치추적 색인

* 이 논문은 교육인적자원부 지방연구중심대학육성사업 (차세대물류 IT 기술연구사업단)의 지원에 의하여 연구되었음.

은 이동체 색인과 달리 태그 식별자(tid), 리더 식별자(rid), 시간(time)으로 구성된 3 차원 도메인에서 모델링된 태그 객체를 저장한다 [1].

RFID 환경에서 태그 위치 색인의 rid 와 tid 도메인은 크기차가 크다. 또한 한쪽 도메인으로 불균등한 영역 질의의 발생이 빈번하다. 따라서, 최소 면적 확장 정책에 따른 기존의 이동체 색인의 삽입 기법을 태그 위치 색인에 그대로 적용했을 경우 질의 영역과 생성된 노드 간의 겹침이 심하게 되어 색인 검색 비용이 증가하는 문제가 발생한다. 이러한 데이터 모델의 변경으로 인한 문제를 해결하기 위해 R*-tree 의 삽입 기법을 기본적으로 따르는 TPIR-tree 의 삽입 방법을 개선한 불균형 확장 정책을 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련연구를 소개하고, 3 장에서는 대상 환경 및 기존 삽입 방법의 문제점을 정의한다. 4 장에서는 삽입 시 도메인의 비율차를 고려한 불균형 확장 정책을 제시한다. 5 장에서는 실험결과에 대해 살펴보고 마지막으로 6 장에서 결론 및 향후 연구를 기술한다.

2. 관련 연구

이 장에서는 기존 이동체 색인의 근간을 이루는 R*-Tree, 3DR-Tree[3], 시간 기반 R-Tree[4]등과 RFID 환경에 적합한 R*-Tree 를 기반으로 한 TPIR-Tree 대하여 기술한다.

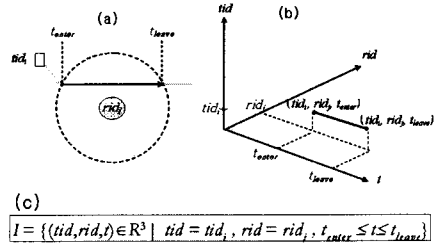
R*-Tree 의 경우 R-tree 가 삽입 시에 영역(area)만을 고려하는 단점을 보완하기 위하여 겹침(overlap)과 가장자리(margin)를 추가적으로 고려한 색인구조이다. 그러나 이동체 색인의 도메인인 x, y 의 경우 동일한 유클리디언 공간(Euclidean space)이므로 도메인의 크기 차이에 대한 고려가 없다.

기존 2 차원 공간 데이터를 저장하는 R-Tree 에 기반한 3DR-Tree 는 시간을 또 다른 하나의 축으로 고려한다. 3DR-Tree 의 경우 두 끝점이 알려져 있는 선분만을 저장할 수 있어 현재 질의를 할 수 없으며 시간 도메인에 대한 고려가 없기 때문에 공간 활용도가 떨어지는 단점이 있다. 또한 현재 시간을 기준으로 객체의 삽입이 진행되기 때문에 시간 축 분할 시 균등 분할하게 되면 과거 노드로의 삽입이 더 이상 없으므로 공간 활용률이 떨어지게 된다.

시간 기반 R-Tree 의 경우 시간 도메인의 성장을 고려하여 시간축 분할 시 비균등 분할 정책을 사용하여 공간 활용도를 높였다. 노드간의 중복과 사장 공간을 최소화 하기 위하여 강제 합병 정책을 사용하여 중복이 심한 노드를 강제 합병 시킨다. 또한 오버플로우 노드의 분할 시에 노드간의 중복을 심하게 하는 원인이 되는 긴 선분을 절단 정책을 사용하여 분할 노드간의 중복을 제거하여 성능을 높이고 있다. 하지만, 이 색인의 경우 이동체가 주기적으로 위치정보를 보고하고 이동체의 분포는 공간상에 균등 분포한다는 가정이 있으므로, 태그 객체의 위치보고가 비주기적이고 태그 객체의 분포는 공간상에서 비 균등 분포를 가지기 때문에 RFID 환경에 그대로 적용할 수가 없다.

3. 대상 환경 및 문제 정의

3.1 대상환경



(그림1) (a) 태그 이벤트 (b) TPIR-Tree 의 태그 객체의 위치 모델 (c)시간 간격 인터벌 I 의 정의

그림 1 의 (a)와 같이 태그 이벤트는 태그가 어떤 시간에 리더에 인식되었을 경우 (tid, rid, t)의 형태로 TPIR-tree 에 저장된다. 이러한 태그 이벤트는 그림 1 의 (b)와 같이 3 차원 공간상에 인터벌로 표현되며 그림 1 의 (c)와 같이 시간 간격 인터벌 I 로 표현된다[1].

3.2 문제정의

기존 이동체 색인을 구성하는 데이터 모델은 공간(x, y)과 시간(time) 도메인이다. RFID 환경에 이동체 색인의 방법을 적용한 TPIR-tree 의 경우 태그 식별자(tid)에 대한 질의가 중요해 tid 가 하나의 축으로 추가되고 공간 도메인(x, y)은 리더 식별자(rid) 축으로 사상된다.

이동체의 공간 도메인인 x, y 는 동일한 유클리디언 공간(Euclidean space)으로 최대 도메인의 크기가 같은데 비해 RFID 도메인인 tid, rid 의 경우 최대 도메인의 크기 차이가 크다. tid 의 경우 EPC 로 이루어져 있으며 전체 도메인의 크기는 EPC(Electronic Product Code)의 종류에 따라 2⁹⁶ 또는 2¹²⁸이다. 반면 rid 의 경우 논리적 리더(Logical reader)로 정의되며 최대 도메인의 크기는 tid 에 비해 상대적으로 적은 크기이다. 예를 들어 1m x 1m 크기의 리더를 지구전체에 펼쳐 놓아도 2⁵⁰ 을 넘지 못한다. 실제 환경에서 리더의 개수는 훨씬 더 적은 수가 될 것이다.

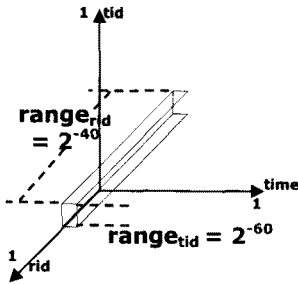
다음과 같은 “100.100.*의 tid 가 오전 11:00~12:00 시 사이에 1~1024 의 rid 중 어떤 rid 에 존재했는가?” 라는 질의가 있을 때 tid 와 rid 도메인의 질의영역을 표현하면 다음과 같다.

위의 질의 예는 도메인 별로 표 1 과 같이 불균등한 질의 영역을 가진다.

<표1> tid, rid 의 불균등 영역 질의 예

	전체도메인	질의범위	정규화된 공간상의 질의 영역
tid	2 ⁹⁶	2 ³⁶ (100.100.*)	2 ³⁶ /2 ⁹⁶ = 2 ⁻⁶⁰
rid	2 ⁵⁰	2 ¹⁰ (1024)	2 ¹⁰ /2 ⁵⁰ = 2 ⁻⁴⁰

이 때 이러한 도메인을 비교하기 위해 각 도메인을 축을 [0,1]³ 의 정규화된 공간(Normalized space)에 표현하면 표 1 과 같이 불균등한 질의 영역으로 표현 된다.



(그림2) RFID 환경의 불균등 질의 예

그림 2 의 예에서 $[0,1]^3$ 으로 정규화된 공간상에서는 오히려 rid 영역이 tid 영역에 비해 2^{20} 배 크다는 것을 알 수 있다. 즉 위의 그림 2 처럼 질의 영역이 rid 축으로 긴 직육면체 모양이라고 할 수 있다.

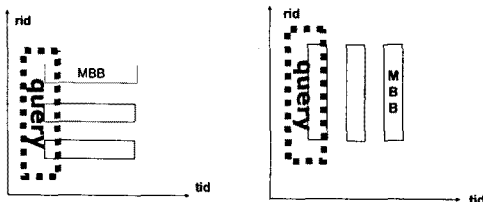
기존 TPIR-Tree 의 삽입 정책은 R*-tree 의 삽입기법을 기본적으로 따르고 있으나 RFID 환경의 위와 같은 tid 및 rid 도메인의 상이한 특성을 고려하지 않아 삽입 시 MBR 과 질의 영역의 겹침이 심해져 노드 접근 횟수가 많아 지는 문제점이 있다. 본 논문의 뒷장에서 상이한 특성을 갖는 두 도메인의 특징을 이용해 삽입 시 불균등 확장 정책을 제안한다.

4. 삽입 정책

이 장에서는 tid, rid 두 도메인의 크기 차이에 의해 발생하는 불균등한 질의 영역의 영향을 기술하고 그 영향에 따라 질의 영역과 MBR 의 겹침을 최소화 할 수 있는 삽입 정책을 제시한다.

4.1 기본 접근 방법

그림 3 과 같이 질의 영역이 tid 에 비해 rid 에 대해 긴 영역일 경우 MBR 의 모양에 따라 노드 접근 횟수가 달라진다. 아래 그림 3 은 쉬운 비교를 위해서 그림 3 의 (a)와 같이 rid 방향으로 긴 MBR 과 그림 3 의 (b)와 같이 tid 방향으로 긴 MBR 의 두 가지 경우에 대해 표현하였다. 질의 영역이 rid 방향으로 긴 모양일 때 그림 3 의 (a)의 MBR 의 경우 그림 3 의 (b)의 MBR 에 비해 질의 시 노드 접근 횟수가 더 많아 진다.



(그림3) (a) 영역질의와 답은풀이 아닌 MBR (b) 영역 질의와 답은풀이 MBR

위의 그림 3 과 같이 질의 모양과 답은풀이 MBR 이 같은 면적을 갖더라도 질의와 겹침 확률이 낮은 것을

알 수 있다. 삽입 시 이러한 특성을 기본적인 접근 방법으로 사용한다.

4.2 ChooseSubtree

R*-tree 의 삽입의 핵심 알고리즘인 ChooseSubtree 를 아래 그림 3 과 같이 변경하였다. 기존 삽입은 디렉토리 노드를 선택할 때 면적 확장값이 가장 적은 쪽 서브트리를 선택하는 Least Area Enlargement 방법을 사용한다. 이 방법을 RFID 환경에 그대로 적용했을 경우 한 축으로 긴 MBR 을 구별 할 수가 없다. 따라서 MBR 을 구성하는 각 축의 변의 길이에 Weight 를 부여하고 둘레길이(margin)을 이용하는 불균등 확장 정책(Disproportional Enlargement Policy)을 사용한다. Weight 는 질의영역의 tid 와 rid 비율을 이용해 실험으로 구할 수 있다.

(알고리즘 1) ChooseSubtree

Algorithm of ChooseSubtree

```

CS1 Set N to be the root
CS2 If [Leaf check]
    N is a leaf,
    return N
    else [determine the minimum overlap cost]
    If the child pointers in N point to leaves, choose the entry in N whose
    rectangle needs least overlap enlargement to include the new data
    rectangle. Resolve ties by choosing the entry whose rectangle needs
    least area enlargement,
    then
    the entry with the rectangle of smallest area
    if [determine the minimum Disproportional Enlargement cost]
    the child pointers in N do not point to leaves, choose the entry in N
    whose rectangle needs Disproportional Enlargement to include the
    new data rectangle. Resolve ties by choosing the entry with the
    rectangle of smallest area
    end
CS3 Set N to be the child node pointed to by the child pointers of the chosen
entry and repeat from CS2
    
```

4.3 불균등 확장 정책(Disproportional Enlargement Policy)

삽입 시 질의 영역의 모양과 유사한 답은풀 MBR 을 선택하기 위해 위한 방법이다. 아래 그림 4 와 같이 새로 삽입된 데이터를 삽입할 MBR 을 선택하기 위해 가중치(weight)를 부여한 각 축의 변의 길이의 합인 MBR 의 둘레길이(margin) 즉 가중치를 부여한 둘레길이의 확장값이 적은 MBR 을 선택하게 된다. 가중치는 불균등한 질의 영역의 비율을 이용하여 실험적으로 구할 수 있다.

수식1 Weighted margin

$$WeightedMargin(mbr, d) = \sum_{j=1}^d Weight_j \times Length_j$$

WeightedMargin 을 수식 1 과 같이 표현할 때 각 mbr 의 WeightedMargin 의 확장값은 수식 2 와 같다.

수식2 Weighted margin enlargement

$$WeightedMargin(MBR, d) - WeightedMargin(mbr, d)$$

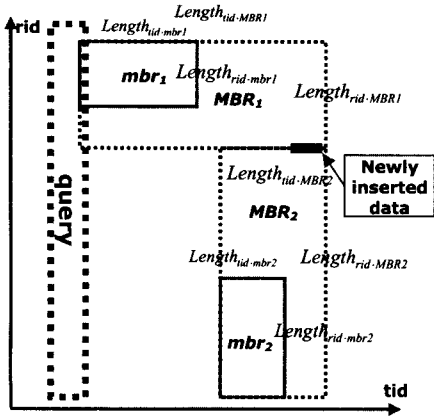
그러므로 그림 4 의 각 MBR 의 Weighted margin enlargement 를 수식 2 로 표현하면 다음과 같다.

Weighted margin enlargement of mbr1

$$= WeightedMargin(MBR1,3) - WeightedMargin(mbr1,3)$$

Weighted margin enlargement of mbr2

$$= WeightedMargin(MBR2,3) - WeightedMargin(mbr2,3)$$



(그림4) 데이터 삽입시 ChooseSubtree 를 위한 변수

다음으로 그림 4 의 예를 사용하여 실제 삽입 시 MBR 을 선택하기 위해 **WeightedMargin** 을 구하는 방법을 설명하였다.

$Length_{tid} = Length_{tid-MBR} - Length_{tid-mbr}$ 이라고 정의하고 $Weight_{tid} = 1$, $Weight_{rid} = 0.1$ 이라고 가정하면 각 mbr 의 **WeightedMargin** 은 다음 표 2 와 같다.

<표2> 각 MBR 별 Weighted margin 의 값계산

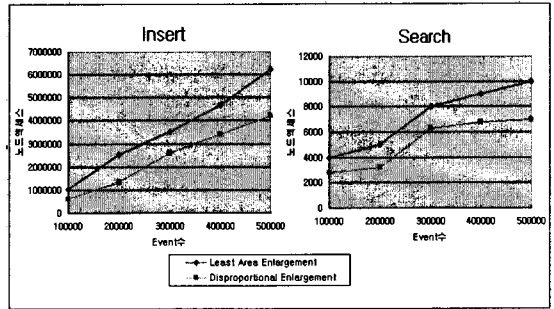
	$Length_{tid}$	$Length_{rid}$	WeightedMargin
mbr1	2	4	$2 \times 0.1 + 4 \times 1 = 4.2$
mbr2	4	2	$4 \times 0.1 + 2 \times 1 = 2.4$

mbr1 의 **WeightedMargin** 의 확장값은 4.2 이고 mbr2 의 **WeightedMargin** 의 확장값은 2.4 이므로 더 적은 값이 확장된 mbr2 가 삽입할 서브트리라 된다.

5. 구현 및 실험 평가

실험 평가에서 기존의 R*-tree 를 기반으로 하는 TPIR-tree 의 삽입 방법인 Least Area Enlargement 방법 과 본 논문에서 제안한 Disproportional Enlargement 방법을 비교하였다. 불균등 영역 질의 크기를 1:10, 1:100, 1:1000 으로 변화시키고 태그 이벤트의 개수를 10 만~50 만개까지 10 만 단위로 실험을 실시했다.

검색 비용은 그림 5 에서 볼 수 있듯이 기존 방법에 비해 20%~40% 가량 성능이 향상된 것을 알 수 있다. 부가적으로 삽입 비용을 비교했을 때 비슷한 성능 향상을 볼 수 있다. 검색 뿐만 아니라 삽입도 성능 향상이 되었다. 이러한 결과가 나오는 이유는 삽입 연산 역시 검색(ChooseSubtree)을 바탕으로 하기 때문이다. 또한 현재 시간의 데이터를 찾아 삽입하는 연산이 필요하다. 이 연산은 극단적인 불균등 영역 질의이기 때문에 성능 향상에 많은 영향을 끼쳤던 것으로 확인된다.



(그림5) 삽입 및 검색 실험 결과

6. 결론 및 향후 연구

이 논문에서는 기존 이동체 색인을 기반으로 RFID 환경의 도메인의 특징에 따른 삽입 방법에 대해서 제시하였다. RFID 환경에서는 도메인의 크기의 차이가 커서 불균등한 질의 영역이 발생하게 된다. 이동체 색인을 기반으로 하는 기존의 RFID 색인의 경우 이러한 특성을 반영하지 못한 삽입정책으로 인해 질의 시에 불균등한 질의 영역과 MBR 의 겹침이 심해져 노드 접근 횟수가 증가하는 문제점이 있었다. 이러한 문제를 해결하기 위해 삽입 시 질의 영역과 답은폴의 MBR 을 선택하는 불균등 확장 정책을 제안하였다. 또한 실험을 통하여 TPIR-tree 에서 기존 삽입 정책인 최소 면적 확장 방법의 삽입 비용과 질의 비용을 비교하였다. 불균등한 질의 영역을 가지는 환경에서 질의 시 20~50%의 성능 향상을 볼 수 있었다. 특히 삽입 비용 또한 검색 비용의 감소로 인해 줄어 드는 효과가 있었다.

향후 연구로써 불균등한 질의 영역에 대해 삽입뿐만 아니라 분할 축선정 및 분할에 대한 연구가 필요하다. 현재 분할 정책은 기존의 Overlap 을 적게 하는 방법을 사용하고 있으며 질의 영역과 답은폴의 MBR 을 선택하는 방법을 Overlap 에 적용하여 Overlap 과 질의영역의 겹침을 줄여서 성능 향상을 시킬 수 있을 것으로 보인다.

참고문헌

- [1] C. H. Ban, B. H. Hong, and D. H. Kim, "Time Parameterized Interval R-tree for Tracing Tags in RFID Systems", 16th Int. Conf. on DEXA, pp. 503-513, 2005.
- [2] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger, "The R*-tree : An Efficient and Robust Access Method for Points and Rectangles", Proc. of the ACM SIGMOD International Conf. on Management of Data, pp.332-331, 1990
- [3] Y. Theodoridis, M. Vassilakopoulos, and T. Sellis, "Spatio-Temporal Indexing for Large Multimedia Applications," In Proc. of the 3rd IEEE Conf. on Multimedia Computing and Systems, pp. 441-448, 1996.
- [4] 전봉기, 홍봉희, "이동체의 색인을 위한 시간 기반 R-트리의 설계 및 구현", 한국정보과학회 논문지 D- 데이터베이스, 30권 3호, pp.320-335, 2003