

공간색인을 이용한 RFID 태그관리 기법

주 현 식*

A RFID Tag Indexing Scheme Using Spatial Index

Heon Sik Joo*

요 약

본 논문은 공간색인을 사용하여 RFID 태그를 관리하는 태그색인 기법을 제안한다. 재고 관리 등에 사용되는 태그는 리더에 의해 위치가 결정된다. 즉, 태그가 부착된 제품이 리더에 인식됨으로써 제품의 위치가 생성되어 추적이 가능하다. 본 논문은 RFID 태그가 부착된 제품을 관리하는 혼합태그색인(hTag-tree: Hybrid Tag index)을 제안한다. hTag-tree는 태그의 특성을 반영하여 빠른 검색이 가능하도록 제안하는 새로운 색인이며 리더의 공간좌표를 이용하여 태그를 관리하는 태그 인덱스이다. 본 제안 색인은 동적환경에서 태그의 삽입, 삭제, 갱신에서 빠른 노드접근이 가능하며, 기존 기법에 비해 태그 검색시 노드접근 횟수를 최소화한다. 또한 기존 태그색인에서 MBR의 확장으로 인하여 조상 노드를 접근함으로써 탐색성능이 저하되는 것을 방지하였다. 제안한 색인의 실험은 태그 인덱스인 Fixed Interval R-tree와 기존의 공간색인인 R-tree를 비교 하였으며 결과적으로 데이터 탐색을 위한 노드접근횟수와 검색 시간에 있어서 hTag-tree가 더 향상된 시간을 나타낸다. 이는 제안 색인을 이용하면 다량의 RFID 태그를 보다 효율적으로 관리할 수 있다는 사실을 보여주고 있는 것이다.

Abstract

This paper proposes a tag indexing scheme for RFID tag using spatial index. The tag being used for the inventory management and the tag's location is determined by the position of readers. Therefore, the reader recognizes the tag, which is attached products and thereby their positions can be traced down. In this paper, we propose hTag-tree(Hybrid Tag index) which manages RFID tag attached products. hTag-tree is a new index, which is based on tag's attributes with fast searching, and this tag index manages RFID tags using reader's location. This tag index accesses rapidly to tags for insertion, deletion and updating in dynamic environment. This can minimize the number of node accesses in tag searching comparing to previous techniques. Also, by the extension of MBR in present tag index, it is helpful to stop the lowering of capacity which can be caused by parent node

• 제1저자 : 주현식

• 투고일 : 2009. 06. 26 심사일 : 2009. 07. 01, 게재확정일 : 2009. 07. 10.

* 삼육대학교 컴퓨터학부 부교수

※ 이 논문은 2008년도 삼육대학교 학술연구비지원에 의해 수행된 연구임

approach. The proposed index experiment deals with the comparison of tag index, Fixed Interval R-tree, and present spatial index, R-tree comparison. As a result, the amount of searching time is significantly shortened through hTag-tree node access in data search. This shows that the use of proposed index improves the capacity of effective management of a large amount of RFID tag.

▶ Keyword : RFID Tag, Reader, Spatial Index, R-tree

I. 서론

유비쿼터스 컴퓨팅의 가장 근본이 되는 기술은 모든 사물을 식별할 수 있는 전파식별(RFID: Radio Frequency IDentification) 기술이다. 특히 RFID는 유비쿼터스 센서 네트워크(USN: Ubiquitous Sensor Network) 환경에서 객체 식별에 응용되는 주요 기술이다[1]. RFID 시스템의 기본 구성은 태그(tag), 리더(reader), RFID 서버(server)로 구성된다. 태그는 객체에 대한 정보를 저장하며, 식별대상 객체에 부착된다. 리더는 태그의 정보를 식별하는 송/수신 기능을 가지며, 인식된 데이터를 서버로 전달한다. 최근 유비쿼터스 센서 네트워크 환경에서 태그는 화물 컨테이너, 동식물 및 각종 상품에 부착되어 기존에 사용되어온 바코드(barcode)를 대체 하고 있으며, RFID 시스템이 확산되고 있다. RFID는 태그를 사용하여 바코드 보다 많은 정보를 저장하여 제품에 부착 및 관리하는 새로운 기술이다. 현재 이러한 RFID 시스템은 다양한 응용분야에서 사용 중이며 향후 이러한 RFID 태그의 증가는 관리의 문제를 야기 시킨다. RFID 시스템은 태그, 리더(reader), 미들웨어(middleware), RFID 서버로 구성되며, 제품 단위로 관리되는 것은 태그이다. 리더는 태그를 읽어 미들웨어에게 전송하며, 인식된 정보는 RFID 서버에서 관리된다.

태그는 자체 전력을 갖는 능동형 태그와 자체 전력이 없는 수동형 태그로 구분된다. 본 연구에서는 수동형 태그를 대상으로 하며, GPS를 갖지 않아 태그는 자신의 위치 값을 갖지 못한다. 그러나 리더(reader)가 태그를 읽음으로써 해당 태그의 위치가 생성되며, 추적이 가능하다. 각 제품에 부착된 수동형 태그들은 제품이 이동되면서 위치가 변경된다. 이러한 특징은 기존의 공간데이터와 유사한 특징을 갖는다. 이동 객체[2,3]는 GPS를 내장하고 있어 자신의 위치정보를 갖는다는 것이 태그와 다르다[4].

공간 데이터를 관리하는 색인은 R-tree, 태그를 관리하는 색인은 FIR-tree가 연구되었다. R-tree는 공간데이터를 MBR(minimum bounding rectangle)로 추상화하여 관리한다. 한 공간의 각 공간데이터들은 MBR로 관리되기 때문에

서로 겹칠 수 있으며, 특히, 검색대상 데이터를 포함하는 두 개 이상의 사각형이 존재할 수 있기 때문에 검색 효율이 저하되는 단점이 있다. 또한 태그를 관리하는 FIR-tree는 R-tree를 활용한 좋은 색인이지만 동적환경에서 태그의 갱신 속도가 현저하게 떨어진다.

본 논문에서는 RFID 태그가 동적 환경에서 리더에 의해 인식됨으로써 위치정보가 자주 변경되는 경우 해당 태그를 빠르게 찾아 정보를 갱신할 수 있도록 해시색인을 R-tree와 함께 사용하여 탐색성능을 향상시킨 새로운 색인을 제안한다. 제안기법의 주요내용(contribution)은 다음과 같다.

- RFID 태그의 이동을 색인화하여 빠른 검색이 가능하다.
- 동적환경에서 태그의 삽입, 삭제, 갱신에서 빠른 노드 접근이 가능하다.
- 태그 검색시 노드접근 횟수를 최소화한다.
- 기존 태그색인에서 MBR의 확장으로 인하여 조상 노드를 접근함으로써 탐색성능이 저하되는 것을 방지하였다.

본 논문의 구성은 다음과 같다. 2장에서는 RFID 환경에서의 태그 인식, 기존 공간 색인인 R-tree에 대해 살펴본다. 3장에서는 제안 기법의 색인 구성을 설명하고 실험평가는 4장에서 보인다. 5장에서는 논문의 결론을 기술하며 논문을 마친다.

II. RFID 태그와 색인

1. RFID 태그와 리더의 인식과정

RFID 시스템은 리더의 인식영역에 들어오는 태그를 인식하여 제품정보를 식별한다. <그림 1>과 같이 태그는 이동 중에 리더의 인식영역에 진입 또는 이탈하게 되며 리더는 읽어 들인 정보를 미들웨어에 전송함으로써 응용 프로그램에서 해당 제품을 검색한다[5,6].

본 논문의 연구 대상인 수동형 태그는 GPS가 없기 때문에 태그 자체에는 위치정보가 없다. 그러나 태그를 인식하는 리더는 특정위치에 고정되어 있어서 위치정보를 가질 수 있다. 따라서 태그는 리

더에 읽힘으로써 위치정보를 가질 수 있기 때문에 공간데이터로 관리 가능하다. 본 논문에서는 위치정보가 없는 태그를 리더의 위치를 활용하여 공간데이터로 관리 한다. <그림 1>과 같이 태그 t는 리더의 인식영역에 들어감으로써 리더의 위치를 갖게 된다[7,8,9].

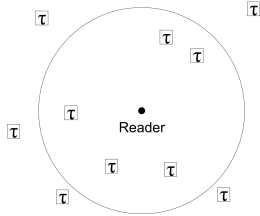


그림 1. RFID 리더의 태그 인식 과정
Fig. 1. Tag interrogation in a RFID reader

2. 공간색인 R-tree

공간 색인은 <그림 2>와 같이 공간 데이터를 최소경계사각형(mbr)으로 추상화 하여 관리한다[2]. 따라서 R-tree는 MBR의 두 대각점의 네 개 좌표를 사용하여 공간 데이터를 관리하며 각 노드는 하위 노드를 모두 포함하는 MBR을 갖는다. 중간 노드의 구조는 <MBR, subnode_pointer>이다. 최하위 단말 노드는 실제 데이터를 가리키는 포인터를 포함한다. R-tree의 구조는 <그림 3>과 같다[10,11].

R-tree는 하나의 데이터가 두 개 이상의 MBR에 포함될 수 있으며, 데이터를 탐색하기 위해서는 상위 노드를 재방문하고 다른 하위노드로 이동하는 과정에서 노드 접근횟수가 증가하는 단점이 있다. 본 논문에서는 이러한 단점을 해결하기 위해 하향식(top-down) 탐색이 아닌 상향식(bottom-up) 탐색을 지원하는 해시 색인을 적용한다[12,13].

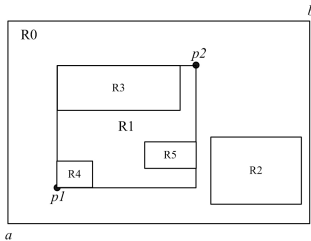


그림 2. 공간 데이터 추상화
Fig. 2. Spatial data approximation

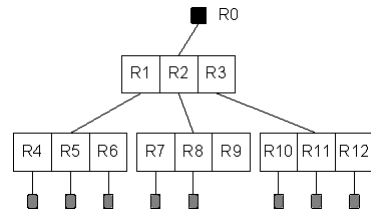


그림 3. R-tree
Fig. 3. R-tree

3. Fixed Interval R-tree

Fixed Interval R-tree(IFR)[14]은 RFID 태그를 관리하는 보기 드문 색인 중 하나이다. IFR은 태그의 위치가 리더에 의해 판단되고 태그의 실제위치는 리더를 벗어날 경우 저장된다는 특징을 사용하여 색인을 구축한다. 특히, 태그가 한 리더에 들어간 후 벗어나지 않은 채 리더에 머물러 있는 경우 태그의 위치가 생성되지 않아 전체 색인을 검색해야하는 단점을 극복한 색인이다.

IFR은 리더에 머물러 있는 태그가 해당 리더를 벗어난 것처럼 관리하기 위해 Leave event에 시간의 최대값(maximum time)을 저장한다. 이렇게 함으로써 태그는 위치와 시간값을 모두 갖게 되며 색인에서 검색이 가능하다. 검색의 종류도 시간값을 중심으로 FIND, LOOK, HISTORY, WITH 질의의 4가지를 지원하며 HISTORY질의를 제외한 나머지 질의에서는 태그의 시간정보가 필요하다.

그러나, FIR에서 최대 시간값으로 태그의 Leave 속성을 저장하는 것은 공간색인에서 태그를 포함하는 상위노드의 MBR(minimum bounding rectangle)을 재구성하도록 하기 때문에 조상노드들의 재구성이 불가피하다. 즉, 많은 노드 접근이 발생하여 전체적인 성능에 영향을 미치게 된다. 또는 현재 태그의 Leave속성에만 저장을 하고 상위노드로 전파하지 않은 경우에는 태그 검색을 할 때 MBR들을 재구성해야하므로 색인의 성능이 저하되는 단점이 있다.

III. 혼합태그색인: hTag index

1. 태그의 공간위치

공간색인인 R-tree에서 탐색 성능 저하는 데이터 탐색과 새로운 데이터를 단말노드에 삽입했을 때 발생한다. 즉, 새로운 데이터가 추가됨으로써 노드의 MBR이 변경되어 영향이 상위노드로 전파되기 때문에 많은 노드 분할과 노드접근이 발

생하여 성능을 저하시킨다. 데이터를 탐색할 때는 루트 노드에서부터 하향식으로 탐색하기 때문에 노드접근 횟수가 증가한다. 예를 들면, MBR R4에 있는 데이터 d를 포함하는 MBR이 R3과 R4이고 서로 겹치면 탐색순서는 상위노드인 R2→R3→R2→R4→d 이다. 즉, 상위노드인 R2를 다시 방문하여 R4로 내려가야 하기 때문에 노드접근횟수가 증가한다. 이는 데이터가 많을수록 더욱 심해진다[15,16,17].

리더에서 인식되는 태그는 특정 리더를 중심으로 공간적 위치를 나타내므로 위치에 대한 밀집도가 높다. 이는 공간 색인에서 MBR의 겹침이 증가하게 되며, 공간색인에서 탐색 성능이 저하된다. 이러한 태그의 특징은 다음의 Lemma와 같이 증명된다. 우리는 Lemma를 대우로 증명하여 참임을 보인다. hTag-tree는 태그가 리더에 진입할 때 리더의 위치를 태그의 위치로 생성 및 저장한다.

Lemma: 태그 Ta가 리더 Ra에서 인식되었다면 태그의 공간 위치정보(Ta.loc)는 리더의 공간위치(Ra.loc)와 같다.

$$Ra \supset Ta \rightarrow Ta.loc = Ra.loc \quad (1)$$

증명: 태그의 공간위치정보가 리더의 공간위치와 같지 않다면 해당 태그는 리더 Ra에서 인식되지 않은 것이다. 즉, 리더는 인식영역에 들어오는 태그들을 인식하며 해당 태그가 어느 리더에 의해 관리되는지 알 수 있다. 그러나 태그의 위치 정보가 다른 리더인 Rb의 위치와 같다면 태그는 Rb에서 인식된 것이며 관리되고 있는 태그이므로 수식(2)는 참이다. 따라서 수식 (1)도 참이다.

$$\sim(Ta.loc = Ra.loc) \rightarrow \sim(Ra \supset Ta) \quad (2)$$

2. 혼합형 태그 색인 알고리즘 및 구조

논문에서 제안하는 혼합형 태그색인은 해시 색인을 함께 사용하여 빠른 검색 성능을 유지한다. 모든 태그의 id는 해시 색인에 등록되며 (tag_id, leafnode_link)의 엔트리로 구성된다. leafnode_link는 R-tree의 단말노드에 직접 연결되어 데이터 탐색 및 갱신을 수행할 때 한 번의 접근으로 태그를 찾을 수 있다.

처음 태그를 읽은 경우 공간 색인을 구성하고 태그id를 해시 색인에 등록한다. 인식된 모든 태그를 동일한 방법으로 두 색인에 등록하여 연결한다. 데이터를 탐색할 때는 공간색인의 루트부터 탐색하지 않고 먼저, 해시 색인에서 찾는다. 탐색된 태그 id의 링크를 통해 단말노드로 이동한다. 만약, 해시 색

인에 태그id가 없다면 새로 등록한다. <그림 4>는 제안 기법에서 공간 데이터를 검색하는 알고리즘이다.

```

1: makeIndex(tag_id, loc_of_reader){
2:   if(isHashed(tag_id) != true){
3:     Link leaf_link =
         insertRtree(tag_id, loc_of_reader);
4:     insertHashIndex(tag_id, leaf_link);
5:   }
6:   return 0;
7: }
8: searchTag(tag_id){
9:   if(isHashed(tag_id) != true)
10:    topDownSearch(tag_id);
11:  else
12:    hashSearch(tag_id);
13: }
    
```

그림 4. 알고리즘
Fig. 4. Algorithm

공간 데이터를 루터를 포함하여 5단계로 구성된 공간색인에서 검색한다면 최소 5번의 노드접근이 필요하다. 단, MBR 사이의 중첩은 없다고 가정한다. 그러나 제안기법인 hTag 색인에서는 해시색인을 경유하므로 한번의 접근으로 태그를 찾는다. 이때, mbr이 중첩되어 있다면 r-tree에서는 수 십번의 노드접근이 발생하게 되지만 hTag는 한 번의 접근횟수를 유지하기 때문에 탐색이 빠르다. <그림 4>는 제안 색인인 hTag와 해시 색인을 생성하고 탐색하는 알고리즘이다.

태그의 위치정보는 R-tree에 MBR로 저장되며 R-tree의 삽입, 삭제 기능을 동일하게 사용하지만 검색은 <그림 5>와 같이 해시 인덱스를 사용한다.

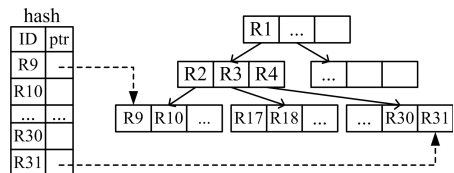


그림 5. 혼합형 태그 색인
Fig. 5. Hybrid tag index

IV. 실험 및 평가

혼합형 태그색인인 hTag는 태그를 탐색하거나 갱신할 때 노드접근횟수와 탐색시간의 두 가지 측면에서 평가하였다. 시험용 데이터 집합은 Data Visual 데이터 생성기를 사용하여 만들었으며 균일한 데이터(uniform dataset, <그림 6>)와 사향 형태의 데이터 집합(skewed dataset, <그림 7>)을 비교하였다. 실험데이터는 균일분포와 사향분포 모두 10,000×10,000의 영역에서 36,000개의 2차원 MBR을 생성하였다. 이 MBR은 태그 데이터이다. 실험환경은 다음 <표 1>과 같다.

표 1. 실험 환경

Table 1. Experimental environment

| 항목 | 내용 |
|-----------------|-----------------------------|
| Uniform dataset | 10,000×10,000 region |
| skewed dataset | 10,000×10,000 region |
| 시뮬레이터 | CSim |
| 하드웨어 | Zeon 3.12Mhz 2CPU, 6GB램 |
| 노드접근횟수 | 탐색을 위해 방문하는 노드 수 |
| 탐색시간 | CPU 시간 |
| 비교 색인 | R-tree, FIR-tree, hTag-tree |

노드접근 횟수는 각 트리에서 태그 데이터를 관리할 때 탐색과 갱신을 위한 접근 횟수의 합으로 나타낸다. 실험결과에서 hTag 색인이 기존의 태그 색인인 FIR-tree나 R-tree보다 낮은 노드 접근횟수를 나타내는데 이는 갱신을 위해 단말노드에 접근할 때 방문해야 하는 노드를 한 번에 찾아가기 때문이다. 이에 반해 R-tree에서는 하향식으로 대상 노드는 검색하기 때문에 중첩된 영역에 있는 데이터에 접근하기 위해 훨씬 많은 노드를 접근하게 된다. 이는 데이터 검색에서도 마찬가지로 결과를 나타낸다.

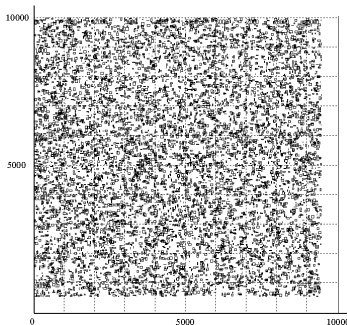


그림 6. 균일 분포 데이터 집합
Fig. 6. Uniform dataset

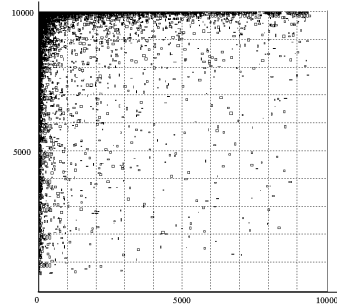


그림 7. 사향분포 데이터 집합
Fig. 7. Skewed dataset

실험은 표준분포 데이터집합과 사향분포 데이터집합에서 대해 각각 탐색시 노드접근횟수와 시간을 측정하였다. 실험에 사용된 질의집합(query set)은 서로 다른 50개의 질의 문으로 구성되며, 각 데이터집합에 적용하여 각 질의 Workload에 대한 평균을 구한 것이다. 데이터의 크기는 5,000개에서 36,000개까지 변화시켜 가면서 실험하였다.

<그림 8>에서 보는 바와 같이 R-tree는 탐색과 갱신에서 노드접근횟수가 FIR-tree보다도 많다. 그러나 hTag-Tree는 갱신을 위해 단말노드에 접근할 때 해시테이블을 경유하므로 접근횟수가 상대적으로 적다. 그러나 노드크기가 20,000일 때 급격히 증가하는데 이것은 단말노드에 접근 후 갱신할 때 상위노드로의 갱신에 대한 반영이 일어나기 때문이다. 즉, MBR을 재구성해야 하기 때문에 Root노드까지의 노드들을 접근하면 분할과 병합이 일어남에 따라 노드접근횟수가 증가하는 것이다. 사향분포의 데이터 집합에서는 <그림 9>에서 보는 것과 같이 5,000개의 태그 데이터에서 FIR-tree, R-tree 그리고 hTag-tree의 경우 모두 균일분포보다 낮은 노드접근횟수를 나타낸다. 결과 적으로 R-tree의 경우 사향분포에서 더 많은 노드접근횟수를 나타낸다. 이는 데이터가 밀집되어 각 데이터들 사이의 중첩(overlap)이 발생하기 때문이다. hTag-tree의 경우도 균일분포에서보다 급격하게 증가한다. FIR-tree는 태그의 leave event를 최대시간으로 관리함으로써 색인에서 태그 검색이 가능하게 하기 때문에 노드 접근이 R-tree보다는 적다.

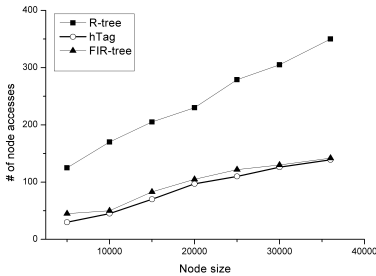


그림 8. 노드접근횟수(균일분포)
Fig. 8. Node accesses the number of times (uniform dataset)

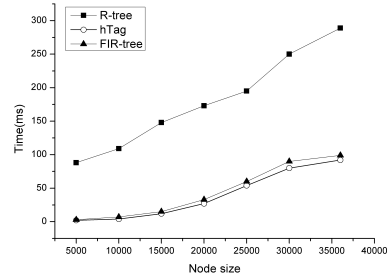


그림 11. 처리시간(사향분포)
Fig. 11. Processing time (skewed dataset)

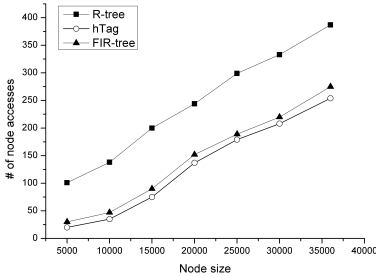


그림 9. 노드접근횟수(사향분포)
Fig. 9. Node accesses the number of times (skewed dataset)

〈그림 10〉과 〈그림 11〉의 처리시간은 동일한 질의집합에 대해 실제 결과가 출력될 때까지의 CPU시간을 측정된 것이다. 처리시간도 hTag-tree가 가장 낮지만 사향 데이터 집합에서는 데이터의 밀집도를 반영하기 때문에 처리시간이 증가한다. 그러나 hTag-tree는 균일한 데이터나 사향분포의 데이터에서 모두 낮은 노드접근횟수와 시간을 나타내기 때문에 성능이 기존 색인들보다 좋다는 것을 알 수 있다.

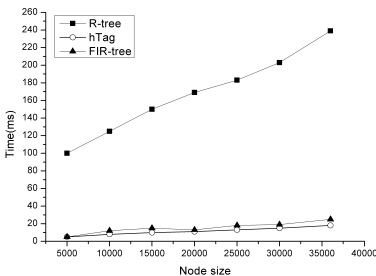


그림 10. 처리시간(균일분포)
Fig. 10. Processing time (uniform dataset)

V. 결론

RFID 시스템에서 태그에 대한 검색, 갱신의 작업은 신속하게 이루어져야한다. 특히 태그를 부착한 제품이 동적으로 이동하는 경우에는 신속성이 더욱 필요하다. 본 논문에서 우리는 태그를 관리하기 위한 색인으로 hTag-tree를 제안하였다. 기존에 태그를 관리하는 색인으로 FIR-tree가 있지만 태그가 리더를 빠져나가는 시간정보를 동일 그룹의 최대시간으로 지정함으로써 상위노드의 MBR을 변경해야 했다. 이는 노드의 분할과 통합이 조상노드들에서 발생하여 노드 접근횟수가 증가함으로써 색인의 성능 저하를 가져오는 단점이 있다.

제안 기법에서는 이러한 성능의 정하 없이 태그 색인에 접근하여 관리할 수 있도록 구성하였다. 결과적으로 hTag-tree는 다음과 같은 장점을 갖는다. 첫째, RFID 태그의 이동을 색인화하여 빠른 검색이 가능하다. 둘째, 동적환경에서 태그의 삽입, 삭제, 갱신에서 빠른 노드접근이 가능하다. 셋째, 태그 검색시 노드접근 횟수를 최소화한다. 마지막으로, 기존 태그색인에서 MBR의 확장으로 인하여 조상 노드를 접근함으로써 탐색성능이 저하되는 것을 방지하였다.

제안기법에서는 태그 색인인 FIR-tree와 공간 색인인 R-tree를 직접 비교하였으며 특히, 제안기법에서는 동적 환경에서 갱신을 위해 빠른 접근을 하도록 해시테이블을 사용하였다. 실험결과에서 보듯이 기존의 공간데이터를 이용한 경우보다 탐색과 갱신에서 hTag-tree가 좋은 성능을 나타낸다. 따라서 본 색인은 향후 태그의 활용이 점차 증가하는 상황에서 태그의 효율적관리를 위해 충분히 응용될 수 있을 것이다.

참고문헌

- [1] K. C. K. Lee, J. Schiffman, B. Zheng, W.-C. Lee, H. V. Leong, "Tracking Nearest Surrounders in Moving Object Environments," Proceedings of the Pervasive Services, 2006 ACS/IEEE International Conference, pp. 3-12, 2006.
- [2] A. Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching," in Proceedings of the ACM SIGMOD International Conference on Management of Data SIGMOD, pp. 47-57, 1984.
- [3] Schilit, B., Adams, N., and Want, R., "Context Aware Computing Application," Processing of the Workshop on Mobile Computing Systems and Application, Santa Cruz, CA, p. 85, 1994.
- [4] I.F.Akyildiz, S. Weilian, Y. Sankarasubramanian, and E. Cayirci, "A Survey on Sensor Networks," Communication Magazine IEEE, Vol. 40, No. 8, pp. 102-114, 2002.
- [5] Dongyu Qiu and Ness B. Shroff, "A Predictive Flow Control Scheme for Efficient Network Utilization and QoS," IEEE/ACM Transactions on Networking, Vol. 12, No. 1, pp. 161, Feb. 2004.
- [6] National Research Council, "Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers," National Academy Press, 2001.
- [7] Park, O.S., Jung, K.R., Kim, S.H., "Location Sensing Tech. and System for Ubiquitous Computing," Weekly Technical Trend, Vol. 1098, pp. 11-21, 2003.
- [8] Weiser, M., "Ubiquitous Computing," ACM Conference on Computer Science, Vol. 26, No. 10, pp. 418-438, 1993.
- [9] H. Knospe and H. Pob, "RFID Security," Information Security Technical Report, Vol. 9, No. 4, pp. 39-50, 2004.
- [10] 양효석, 김병찬, 양대용, "섬유산업의 RFID기술 적용에 관한 연구", 한국컴퓨터정보학회논문지, 제13권, 제4호, 195-204쪽, 2008년 7월.
- [11] 주해중, 박영배, 최창훈, "이동 객체의 실시간 연속 질의를 위한 모바일 클라이언트-서버 시스템," 한국컴퓨터정보학회 논문지, 제11권, 6호, 289-298쪽, 2006년 12월.
- [12] 주현식, "지그비기술을 이용한 무선 기반의 출입통제 시스템 설계 및 구현," 한국컴퓨터학회 논문지, 제13권, 2호, 211-220쪽, 2008년 3월.
- [13] 이기욱, 성창규, "유비쿼터스 센서 네트워크 기반의 상황정보 모니터링 시스템 구현," 한국컴퓨터정보학회 논문지, 제11권, 제5호, 259-265쪽, 2006년 11월.
- [14] S. Ahn, B. Hong, C. Ban, K. Lee, "Design and Implementation of an Index Structure Using Fixed Intervals for Tracing of RFID Tags," ICCSA2006, LNCS3981, pp. 175-185, 2006.
- [15] 최용식, 김성선, 신승호, "유비쿼터스 환경에서 센서 노드의 관리와 망 구성을 위한 RFID 미들웨어 프로토콜에 관한 연구," 한국컴퓨터정보학회 논문지, 제12권, 제3호, 155-163쪽, 2007년 7월.
- [16] 이기영, 김동오, "유비쿼터스 컴퓨팅 환경에서의 위치 데이터 관리시스템의 설계," 한국컴퓨터정보학회 논문지, 제12권, 제6호, 115-121쪽, 2007년 12월.
- [17] 이상렬, "RFID 시스템의 개선된 인증 프로토콜," 한국컴퓨터정보학회논문지, 제12권, 제6호, 193-200쪽, 2007년 12월.

저자소개



주현식

2005년 아주대학교 컴퓨터공학과
(공학박사)

1997년 ~ 현재 : 삼육대학교 컴퓨터
학부 부교수

관심분야: 유비쿼터스, 모바일컴퓨팅,
멀티미디어, 정보보안