

SQL 기반 RFID 클리닝 모듈 설계

윤희성^o, 김동균, 이상정
순천향대학교 컴퓨터학부
e-mail : iloveyr2@gmail.com, kdk70@sch.ac.kr, sjlee@sch.ac.kr,

Design of SQL Based RFID Cleaning Module

Hee-Sung Yun^o, Dong-Kyun Kim, Sang-Jung Lee
Dept. of Computer Science and Engineering, Soonchunhyang University

요 약

RFID 기술의 상용화를 저해하는 한 요소인 태그 인식률 문제를 보완하기 위한 클리닝 모듈을 설계한다. 클리닝 모듈은 RFID 리더로부터 원본 데이터를 클리닝 모듈을 통해 애플리케이션에서 사용할 수 있는 수준의 정보로 가공한다. 클리닝 모듈의 성능을 확인하기 위해 태그의 논리적인 구역을 정하고 태그의 이동을 추적한다. 실험결과를 통해 클리닝 모듈 적용 전후를 비교하여 모듈의 성능을 평가한다.

1. 서론

최근 RFID(Radio Frequency Identification) 기술을 이용한 다양한 애플리케이션이 개발되고 있다[1]. RFID 기술은 태그를 이용하여 사물의 정보와 주변 환경정보를 무선으로 전송하는 비접촉식 인식 시스템이다. RFID 기술은 유비쿼터스의 핵심기술 중의 하나이며 현재 사용되는 많은 시스템들을 대체할 수 있는 기술로 평가 받고 있다. RFID 기술의 적용분야는 물류/유통, 제조, 국방, 의료 등 매우 다양하다. 하지만 RFID 시스템 개발을 저해하는 요소로 태그에 대한 인식률 문제를 가지고 있다. 인식률 보완을 위한 연구가 진행되고 있으며, 문제 해결 방법의 하나로서 데이터 클리닝 기법이 있다. 인식률 향상은 보다 가치 있는 정보를 추출함으로써 RFID 애플리케이션의 성공을 보장한다. 따라서 본 논문에서는 RFID 시스템의 인식률을 향상시키기 위해 SQL 기반의 클리닝 모듈을 설계하고 구현한다. 또한 클리닝 모듈의 성능을 실험을 통해 평가한다[2].

2. RFID 클리닝 단계

객체 혹은 사물에 대한 인식기술은 바코드, OCR (Optical Character Recognition), 생체인식, 스마트 카드, RFID 등이 있다. 그 중 RFID 기술은 다른 인식 시스템들보다 상대적으로 장애물의 영향에 강하며, 빠른 인식 속도, 긴 인식거리 등 다양한 장점을 지니고 있다. 또한 광범위하게 사용되고 있는 바코드를 대체할 수 있는 기술로 인식되고 있다. 하지만 이러한 장점에도 불구하고 RFID 시스템의 인식률 문제는 상용화를 가로막는 중요한 요소이다. 일반적으로 RFID 기술의 인식률 저하의 원인은 다음의 두 가지로 분류된다 [3].

- **False Negatives:** 태그가 실제로는 존재하지만 인식하지 못한 경우를 의미한다. 지속적으로 특정 논리구역에 존재하는 태그가 인식이 되지 않는 경우가 포함된다.
- **False Positives:** 태그가 실제로는 존재하지 않지만 인식을 한 경우이다. 특정 구역에 존재하는 태그가 관련이 없는 구역의 안테나에서 인식되는 중복인식은 False Positive의 대표적인 예이다.

위와 같은 문제들은 RFID 리더의 물리적 배치, 태그 인식 충돌, 환경적인 요인 등이 원인이 되어 발생하게 된다. 이렇게 불완전한 데이터를 직접 애플리케이션에 사용하기에는 무리가 따르며, 불완전성을 보완하기 위해 원본 데이터를 가공하는 여러 기법들이 연구되고 있다. 원본 데이터를 가공하는 기법들은 크게 두 가지로 분류된다. 첫 번째는 전통적인 형태의 데이터 클리닝 기법이고, 두 번째는 수신기 기반 데이터 관리 기법이다[4,5].

RFID 데이터를 분석하기 위해서는 의미 있는 시간적, 공간적 단위가 필요하다. 다음은 시공간적 단위를 설명한다.

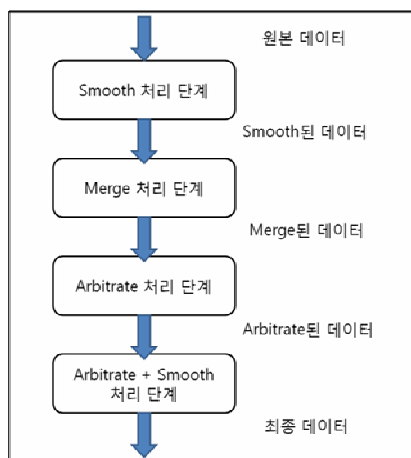
- **시간적 단위:** 애플리케이션에서 데이터가 의미를 지니기 위한 최소한의 시간 단위. 특정 애플리케이션에 따른 시간 단위를 지정하고, 연속적인 시간단위들의 관계를 분석하여 이상유무 발견 시 이를 보정하는데 시간적 단위가 사용된다.
- **공간적 단위:** 애플리케이션에서 데이터가 의미를 지니기 위한 최소한의 공간 단위. 하나의

공간단위를 두 개의 안테나가 중복하여 태그 인식을 수행할 때 근접(중복)그룹으로 지정하여 값을 보정하는데 공간적 단위가 이용된다.

시공간적 단위는 클리닝 모듈 처리단계의 기본 단위로 사용된다. 불완전한 원본 데이터를 보정하는 클리닝의 대표적인 3 가지 기법은 다음과 같다[6].

- **Smooth:** 시간적 단위를 기준으로 태그가 인식된 단위시간당 횟수가 임계값 이상일 경우, 그 공간에 태그가 존재한다고 가정한다. 이 단계에서 False negative 문제를 보정할 수 있다.
- **Merge:** 공간적 단위를 기준으로 근접그룹 내의 둘 이상의 안테나에 대한 태그인식 횟수를 파악하고, 각 안테나의 인식결과를 논리적인 하나의 스트림으로 통합한다. 같은 근접 그룹을 모니터링 하는 둘 이상의 안테나 데이터를 하나로 통합하여 태그 인식 데이터의 신뢰성을 증대시킨다.
- **Arbitrate:** 서로 다른 공간적 단위의 동일한 태그에 대한 중복 데이터가 발생한 경우 각 공간내의 태그 인식 횟수를 비교하여, 큰 값을 지닌 공간에 태그가 존재한다고 가정한다.

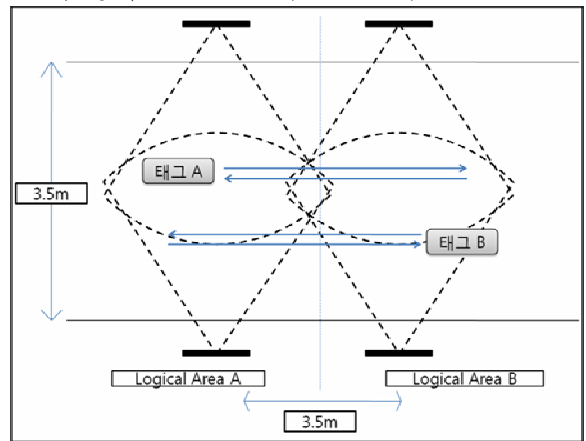
RFID 원본 데이터를 보정하는 클리닝 기법의 대표적인 3 가지 기법(Smooth, Merge, Arbitrate)들은 각각 독립적으로 적용될 수도 있고, 여러 개가 한꺼번에 적용될 수도 있다. 클리닝 기법의 복수 적용 시 각 단계를 어떤 순서로 적용하느냐에 따라 인식률의 성능이 다를 수 있다. 본 논문에서는 그림 1 과 같이 Smooth, Merge, Arbitrate 의 순서로 적용한다. 마지막 단계인 Smooth+Arbitrate 처리 단계는 3 단계의 클리닝 단계를 거친 후에도 남아있는 False Positive 를 보정해주는 역할을 한다.



<그림 1> 클리닝 모듈 처리 단계

SQL 기반 클리닝 모듈은 DB 에 저장된 원본 데이터를 각 SQL 문으로 구성된 몇 가지 단계의 보정작업을 통해 데이터의 신뢰성을 향상시키는 기법이다. 다른 애플리케이션에 적용 시 유사한 형태로 적용이 가능하고, 모듈 변경 시에도 유동적인 적용을 할 수 있다.

SQL 기반 클리닝 모듈 성능을 평가하기 위한 실험은 논문에서 설계하는 클리닝 모듈의 역할을 평가할 수 있는 최소한의 규모로 구현한다. 그림 2 처럼 논리적인 영역 A, B 로 나누어지며 각 구간 사이의 통로 끝에는 각각 2 개의 안테나를 배치한다. 안테나는 정확한 실험을 위해 높이 및 각도를 동일하게 고정한다. 실험에 사용하는 리더의 인식범위는 일반적으로 반지름이 6m 인 부채꼴 형태이다. 안테나 배치 시 안테나 간의 거리를 3.5m 로 설정하여 실제상황보다 데이터 중복인식이 많이 발생하도록 배치하였다. 논리적 영역 A, B 에는 각각 태그를 지닌 사용자가 존재하며, 사용자 A 는 1 분 간격으로 반대편 논리 영역으로 편도 이동을 한다. 또한 사용자 B 는 2 분 간격으로 반대편 논리 영역으로 편도 이동을 한다.



<그림 2> 실험 구성도

리더로부터의 태그 정보와 클리닝 처리후의 정보를 저장하는, 5 개의 테이블로 구성된다. 리더로부터의 태그 정보를 실시간으로 저장하는 Raw_data 테이블, 각 클리닝 단계를 거친 결과를 저장하는 Smooth_res, Merge_res, Arbitrate_res, SA_result 테이블이 있다. Raw_data, Smooth_res 테이블의 형태는 동일하며 Date, Tag_id, Antenna, Read_count 필드를 가진다. Merge_res 테이블은 Antenna 필드를 논리적인 구역으로 매핑시킨 Location 필드를 가진다. Arbitrate_res 테이블은 Merge_res 테이블과 동일한 구조를 가진다. 태그의 위치를 최종적으로 판단한 결과를 지니는 SA_result 테이블은 Merge_res 테이블의 Read_count 를 제외한 Date, Tag_id, Location 필드를 갖는다.

Date(PK)	Tag_id	Antenna	Read_count
2007-09-24 10:11:10	0002	A	2
2007-09-24 10:11:10	0014	B	5
2007-09-24 10:11:11	0002	A	3

<표 1> Raw_data, Smooth_res 테이블

3. SQL 기반 클리닝 모듈

Date(PK)	Tag_id	Location	Read_count
2007-09-24 10:11:10	0002	A	4
2007-09-24 10:11:11	0014	B	7

<표 2> Merge_res, Arbitrate_res 테이블

Date(PK)	Tag_id	Location
2007-09-24 10:11:11	0002	A
2007-09-24 10:11:11	0014	B

<표 3> SA_result 테이블

태그 정보가 데이터베이스에 저장되어 클리닝 모듈에서 처리될 때, 초당 인식횟수 설정값에 관계없이 1회 이상일 경우 태그가 인식된 것으로 가정한다. 그림 3,4,5,6은 클리닝 모듈의 각 단계별 SQL 문을 보여준다.

Smooth 단계를 위한 SQL 문의 예는 그림 3과 같다. 현재 시각부터 2초 전까지의 총 3초간의 인식 데이터에서 2개 이상의 단위시간 구간에서 인식이 있는 경우, 태그가 존재하는 것으로 가정한다. SQL 문의 결과는 태그가 인식된 단위시간의 횟수이다.

```
select Read_count(Tag_id)
from Raw_data where Date <= (select max(Date) from Raw_data)
and Date >= (select max(Date)-2 from Raw_data)
and Tag_id = 1;
```

<그림 3> Smooth 처리를 위한 SQL 문

Merge 단계의 SQL 문은 그림 4에 나타난다. 이 단계에서는 같은 근접그룹을 모니터링하는 복수개의 안테나에 대한 태그 인식정보를 하나로 통합한다. 같은 근접그룹내의 안테나 A와 B 중 하나라도 태그 X를 인식한 경우 그 근접그룹 내에 태그 X가 존재하는 것으로 가정한다. SQL 문의 결과는 현재시각에 같은 근접그룹 내에서 태그가 인식된 안테나 번호이다.

```
select Antenna
from Smooth_res where Date = (select max(Date) from Smooth_res)
and Tag_id = 1;
```

<그림 4> Merge 처리를 위한 SQL 문

Arbitrate 단계의 SQL 문은 그림 5와 같다. 하나의 태그가 중복인식이 발생하였을 경우 공간별로 현재시각부터 2초 전까지의 인식횟수의 합을 구한다. 이 결과가 큰 쪽에 태그가 존재하는 것으로 가정한다. SQL 문의 결과는 각 안테나별 실제 인식횟수의 합이다.

```
select Location, sum(Read_count)
from Merge_res where Date >= (select max(Date) - 2 from Merge_res)
group by Location;
```

<그림 5> Arbitrate 처리를 위한 SQL 문

클리닝 모듈의 마지막 단계는 Smooth+Arbitrate 처리이다. 이 단계의 SQL 문은 그림 6이며, 이전의 클리닝 단계에서 처리되지 못한 오류들을 마지막으로 보정하게 된다. Arbitrate 단계에서 가중치를 기반으로 한 중복인식 보정시 가중치가 동일할 경우 여전히 중복인식이 나타나게 된다. 이 때 Smooth+Arbitrate 처리는

각 공간별 태그인식의 단위시간당 횟수를 계산한다. 이 결과를 토대로 큰 값을 가지는 구역에 태그가 있는 것으로 가정한다[7]. SQL 문의 결과는 각 구역의 단위시간당 인식횟수의 합이다.

```
select Tag_id, count(Tag_id)
from Arbitrate_res where Date >= (select max(Date) - 2 from Arbitrate_res)
group by Tag_id;
```

<그림 6> Smooth+Arbitrate 처리를 위한 SQL 문

4. 실험 및 성능 평가

4.1 실험 환경

실험에 사용된 RFID 리더는 4개의 ALR 9611BC 서클러 안테나를 포함하는 ALIEN사의 ALR-9800이며 900Mhz 대의 주파수를 사용한다[8]. 태그는 EPC Class 1, Gen2 규약을 따르는 ALIEN사의 ALL-9460 "Omni-Squiggle"을 사용한다. 추가적으로 안테나가 태그를 인식하기 위한 전파 발산 시기가 겹치는 경우 신호 감쇄가 일어나 성능저하가 발생할 수 있기 때문에 안테나는 순차적으로 신호를 발생하도록 설정되었다. 리더의 read rate는 10Hz로 설정하였으며, 수신된 태그 데이터를 TCP/IP를 통해 서버로 전송하도록 notify 모드를 설정하였다. 이외의 기타 리더설정은 장치 기본으로 한다. 데이터베이스는 MySQL을 사용한다[9].

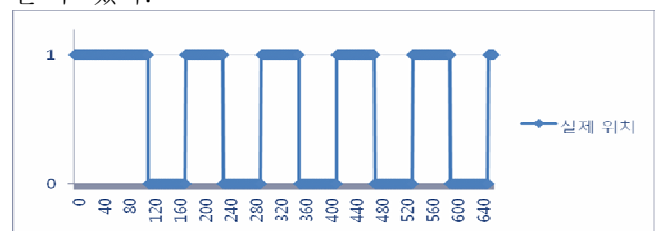
4.2 성능 분석 및 평가

실험결과의 인식률을 측정 한 방법은 인식된 구간(단위: 초)을 전체 인식 구간으로 나눈 후 백분율로 나타내었다. 이 때, 두 개의 안테나 중 하나의 안테나에만 인식된 경우라도 인식이 된 것으로 취급한다. 이렇게 계산한 태그 인식률은 표 4와 같다.

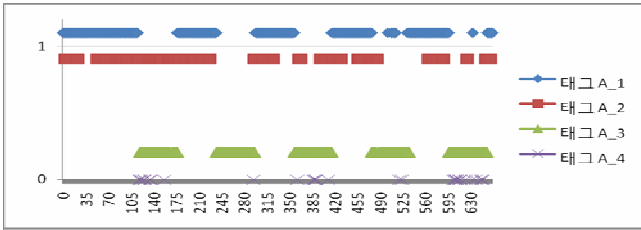
	태그 A	태그 B	평균
인식률	86.2%	74.1%	80.15%

<표 4> 실험 인식률 표

실험의 결과는 주변환경의 작은 변화에도 영향을 받으며, 동일한 조건하에 실험을 실시해도 다른 인식률 값을 얻을 수 있다. 표 4에 나타난 인식률이 일반적인 환경에서의 인식률보다 높게 나타난 것은 실험 환경 자체의 이상적인 조건 때문이라고 추론할 수 있다. 실험 결과에서 발생하는 오류들을 살펴보면 첫 번째로 False negative로서 전체 실험에 전반적으로 특정 논리구역에 태그가 존재하지만 인식되지 않는 오류가 발견되었다. 두 번째로 False positive로서 그림 8의 140초 구간을 살펴보면 중복인식이 발생하였음을 알 수 있다.



<그림 7> 태그 A의 실제위치



<그림 8> 태그 A의 인식결과

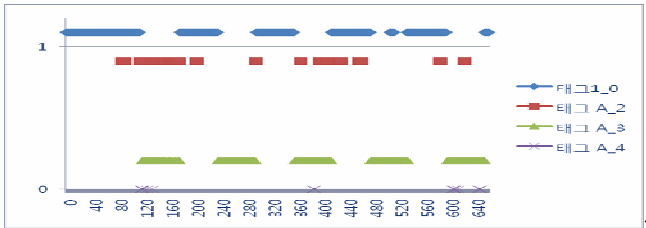
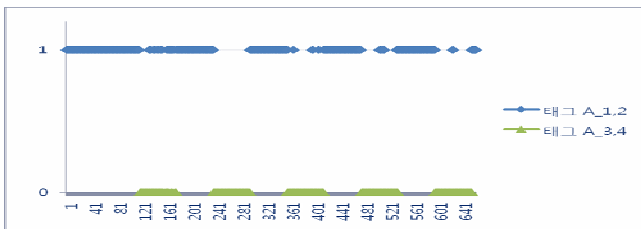


그림 9> Smooth 처리단계 후의 그래프



<그림 10> Merge 처리단계 후의 그래프

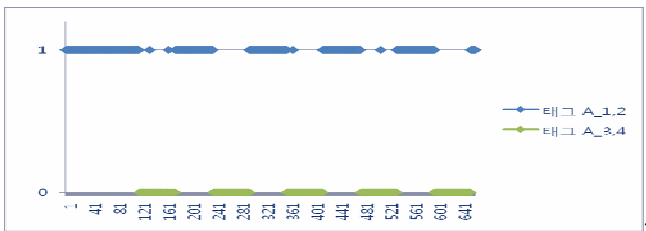


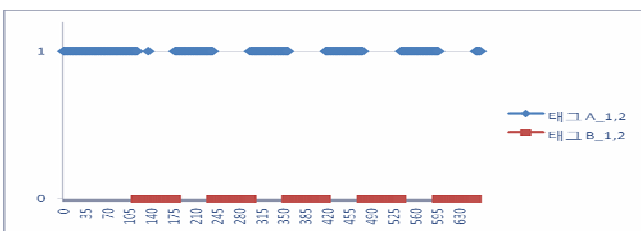
그림 11> Arbitrate 처리단계 후의 그래프

클리닝 모듈의 Arbitrate 단계까지 거친 후의 인식률은 표 5와 같다.

	태그 A	태그 B	평균
인식률	97.3%	98.6%	97.95%

<표 5> Arbitrate 처리단계 후의 인식률

3 단계의 클리닝을 거친 인식률은 상당히 향상되었으나, 그림 12에서 나타나듯이 논리적으로 불가능한 중복인식이 아직도 남아있음을 확인할 수 있다. 이러한 문제를 최종적으로 해결하는 Smooth+Arbitrate 단계를 거친 후의 그래프 및 인식률은 다음과 같다.



<그림 12> Smooth+Arbitrate 처리단계 후의 그래프

	태그 A	태그 B	평균
인식률	98.5%	99.1%	98.8%

<표 6> Smooth+Arbitrate 처리단계 후의 인식률

Smooth+Arbitrate 처리 단계를 거친 후의 그래프를 살펴보면, Arbitrate 단계 이후에도 나타나던 중복인식 부분이 대부분 사라진 것을 확인할 수 있다.

중복을 의도적으로 발생시킨 것을 제외하고 실험의 환경이나 여러 조건들은 상당히 이상적인 상태에서 실시되었다. 그럼에도 불구하고 실제 인식률 향상은 상당하였고, 태그 인식률은 태그 A에서 12.3%, 태그 B에서 24.7%가 향상되었다.

5. 결론

본 논문에서는 SQL 기반의 RFID 클리닝 모듈을 설계, 구현하였다. 실험 결과는 다양한 주변 환경 및 배치에 따른 작은 차이에도 다른 결과를 나타낼 수 있다. 실험을 통해 SQL 기반 클리닝 모듈이 RFID 시스템의 고질적인 문제인 인식률을 상당히 향상시키는 것을 확인하였다. 클리닝 모듈은 다른 시스템에 적용 시 어느 정도의 변형 테스트가 필요하며, 유사한 양식으로 임계값이나 SQL 문의 설정 값 등을 수정하여 쉽게 확장 적용이 가능한 장점을 가진다.

참고문헌

- [1]C. Floerkemeier et al.. Issues with RFID usage in ubiquitous computing applications. In A. Ferscha et al., eds., Pervasive Computing: Second International Conference, PERVASIVE2004. 2004.
- [2] H.Galhardas, et al.. Declarative data cleaning: Language, model, and algorithms. In VLDB, pp. 371-380. 2001.
- [3]Ahmad Rahmati, Lin Zhong Dep. of ECE, Rice University. Reliability Techniques for RFID-Based Object Tracking Applications
- [4]E. Rahm et al.. Data cleaning: Problems and current approaches. IEEE Data Eng. Bull., 23(4):3-13,2000
- [5]Anish Das Sarma, Shawn R. Jeffery, Michael J. Franklin, Jennifer Widom. Estimating Data Stream Quality for Object-Detection Applications. Technical Report No. UCB/ECS-2005-23
- [6]S. R. Heffrey, G. Alonso, M. Franklin, W. Hong, and J. Widom. A pipelined framework for online cleaning of sensor data streams. In Proc. 2006 Int. Conf. Data Engineering (ICDE'06), Atlanta, Georgia, April 2006
- [7] S. R. Jeffery, M. Garofalakis, and M. J. Franklin. Adaptive Cleaning for RFID Data Streams. Under submission to SIGMOD, 2006.
- [8]Alien ALR-9800 915MHZ RFID Reader. <http://www.alientechnology.com/readers/alr9800.php>
- [9]MySQL, <http://www.mysql.com>