RFID 스트리밍 데이터 처리를 위한 연속 질의의 변환 기법

박 재 관[†] · 홍 봉 희[††] · 반 재 훈[†††]

요 약

RFID 미들웨어 시스템은 애플리케이션의 질의를 처리하기 위해서 다수의 RFID 판독기에 의해 지속적으로 인식되는 RFID 스트리밍 데이터를 수집 및 정제한다. 이러한 질의들은 미들웨어에서 일정기간 동안 수행되기 때문에 연속 질의라고 불린다. 미들웨어의 성능을 개선하기 위해서는, 이러한 연속 질의를 효과적으로 처리하기 위한 색인이 필수적이다. 최근, 데이터가 아닌 질의를 기반으로 색인을 구축하는 질의 색인 기법들이 제안되었으며 이러한 기법들은 이동체 위치 스트리밍 데이터 혹은 센서 스트리밍 데이터에 대하여 연속 질의를 처리하는 환경에서 향상된 성능을 보여준다. EPCglobal은 RFID 애플리케이션을 위한 표준 질의 인터페이스인 Event Cycle Specification (ECSpec)을 제안하였다. ECSpec 기반의 연속 질의는 질의의 조건을 대상 도메인 공간에 표현하면 다수의 세그먼트로 표현되는 특징이 있다. 이러한 RFID 질의 색인의 데이터에 대하여 기존의 질의 색인을 사용하면 다수의 데이터를 삽입해야 하기 때문에 색인을 구축 및 유지하기 위한 비용이 커지게 된다. 이러한 문제를 해결하기 위해서, 이 논문에서는 다수의 세그먼트를 새로운 변환공간으로 표현하고 이것을 결집하여 단일 데이터로써 표현하는 결집 변환(Aggregate Transformation) 기법을 제안한다. 또한, 기존 질의 색인과 변환 기법을 적용한 색인의 성능을 비교한다.

키워드 : 스트림 데이터 처리, 연속 질의, 질의 색인

A Transformation Scheme for Continuous Queries on RFID Streaming Data

Jaekwan Park[†] · Bonghee Hong[††] · Chaehoon Ban[†††]

ABSTRACT

RFID middleware systems collect and filter the RFID streaming data gathered continuously by numerous readers in order to process requests from applications. These requests are called continuous queries because they are kept on executing during certain periods. To enhance the performance of the middleware, it is required to build an index to process the continuous queries efficiently. Several approaches of building an index on not data records but queries, called *Query Index*, are proposed and widely used for evaluating continuous queries over streaming data. The EPCglobal proposed an *Event Cycle Specification* (*ECSpec*) model, which is a standard query interface for RFID applications. Continuous queries based on ECSpec consist of a large number of segments for representing the query conditions. The problem with using any of the existing query indexes on these continuous queries is that it takes a long time to build the index because it is necessary to insert a large number of segments into the index. To solve this problem, we propose an *Aggregate Transformation* that converts a group of segments into a compressed data which is representative of the segments. We compare the performance of a transformed index with the existing query indexes.

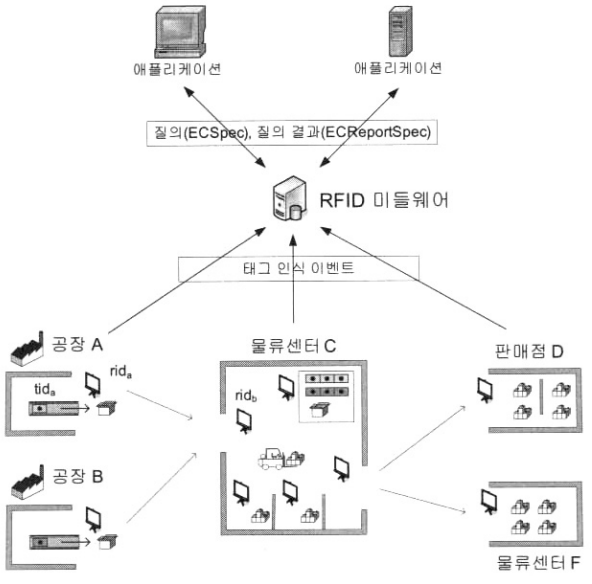
Key Words : Stream Data Process, Continuous Query, Query Index

1. 서 론

Radio frequency identification (RFID) 기술은 자동 객체 식별 및 객체 추적을 위한 효과적인 솔루션으로써 각광받고 있다. RFID 시스템은 물리적인 추적 대상 객체에 부착하는 전자 태그와 태그를 인식하기 위한 판독기, 그리고 애플리케이션의 요구사항에 따라 판독기에서 수집되는 RFID 데이터를 처리하는 미들웨어로 구성된다. RFID 시스템의 애플리케이션은 자동 식별, 자산 추적 그리고 공급망 관리 등 다양하다. 이러한 애플리케이션에서 RFID 데이터는 다수의 판독기에서 계속적으로 수집되는 스트리밍 데이터이다[5, 15].

RFID 미들웨어는 다음과 같은 2가지 특징을 가진다. 첫째, RFID 미들웨어는 판독기에서 전자태그를 인식하면서 끊임없이 발생하는 이벤트를 수집한다. 예를 들면, 그림 1과 같이 공장 A에서 생산된 제품(tid_a)이 판독기(rid_a)를 거쳐 출고될 때 또는 물류센터 C의 판독기(rid_b)를 통해 입고될 때 등 물류 흐름 곳곳에서 RFID 데이터가 계속적으로 발생한다. 둘째, RFID 미들웨어는 각 애플리케이션이 원하는 전자태그

※ 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음
† 준 회 원 : 부산대학교 컴퓨터공학과 박사과정
†† 정 회 원 : 부산대학교 컴퓨터공학과 교수
††† 정 회 원 : 경남정보대학 인터넷응용계열 교수
논문접수 : 2006년 11월 29일, 심사완료 : 2007년 4월 9일



(그림 1) RFID 미들웨어 시스템

정보를 명시한 질의를 등록할 수 있도록 인터페이스를 제공하며 이를 관리한다. 이 질의는 애플리케이션이 어떤 곳에서 인식한, 어떤 태그 정보를 원하는지 명시하는 것으로 판독기 정보와 전자 태그 정보로 표현된다. 예를 들면, 그림 1에서와 같이 다수의 애플리케이션이 미들웨어로 질의(Event Cycle Specification, ECSpec)를 등록하면 미들웨어는 이에 해당하는 전자태그 정보를 일정기간 수집한 후 질의 결과(Event Cycle Report Specification, ECReportSpec)를 애플리케이션에 전달한다.

RFID 미들웨어는 애플리케이션의 요구사항, 즉 질의를 처리하기 위해서 RFID 판독기에서 인식되는 RFID 데이터를 수집 및 정제한다. 이러한 질의는 미들웨어에서 일정기간 동안 계속해서 수행되기 때문에 연속 질의(continuous query)라고 불린다. 따라서, 미들웨어는 연속 질의를 효과적으로 처리하기 위한 색인이 필수적이다. 최근, 연속 질의를 데이터로 고려하여 색인을 구축하는 접근 방법인 질의 색인 기법이 제안되었다. 이 기법은 스트리밍 데이터 환경에서 전통적인 데이터 색인을 사용할 경우 발생하는 지속적인 색인 갱신 및 질의 처리 성능 저하 문제를 해결한다. 따라서, RFID 미들웨어의 성능 향상을 위해서 질의 색인을 도입하는 것이 적합하지만, 질의 색인의 데이터인 연속 질의가 기존 연구에서 다루었던 연속 질의와는 다른 특징이 존재하므로 이를 고려한 질의 색인이 필요하다.

CQI[3]와 VCR[8]은 이동체 위치 스트리밍 데이터에 대하여 연속 질의를 효과적으로 처리하기 위한 질의 색인이다. 이 기법들은 영역 질의를 고정 크기의 격자들로 분할하고 분할에 사용된 격자들을 질의 색인에 삽입하는 방법이다. 탐색 과정은 이동체가 보고하는 위치를 포함하는 격자들을 찾아서 해당 영역 질의를 추출하는 과정으로 구성된다. NiagaraCQ[6]와 TelegraphCQ[10]와 같은 센서 데이터 스트림 시스템에서는 IBS-tree를 질의 색인으로 사용한다. 이 색인은 질의의 predicate을 만족하는 간격을 저장하고 특정 센서 노드에서 감지 이벤트가 발생할 때 감지된 센싱값을 포함하는 질의들을 찾는다.

RFID 시스템에 대한 표준화 기구인 EPCglobal은 RFID 애플리케이션을 위한 표준 질의 인터페이스인 ECSpec(Event Cycle Specification)[1]을 제안하였다. 이 스펙은 다양한 질의를 지원하기 위한 인자로써 중요한 두 가지 predicates를 포함하고 있는데, 이 predicates은 판독기와 태그에 대한 필터링 조건으로 구성된다. ECSpec의 예를들면 *"readerID=1~3, EPC pattern=<10.[1-2].[3001-4000]>"*와 같은 형태로 표현된다. 첫번째 predicate은 판독기 식별자의 간격이며 두번째 predicate은 태그의 필터링 조건을 표현하는 방식인 *EPC (Electronic Product Code) Pattern*이다. EPC Pattern은 패턴 형태로 표현되기 때문에 태그 식별자의 다수 간격들로 표현된다.

표준을 따르기 위해서, RFID 질의 색인은 ECSpec의 predicates을 표현하는 연속 질의를 기반으로 구축되어야 한다. 이러한 경우, 연속 질의는 판독기 식별자 도메인(Reader Identification Domain, RID)에서의 간격과 태그 식별자 도메인(Tag Identification Domain, TID)에서의 다수의 간격들로 구성되기 때문에 2차원 색인 공간(RID, TID)에서 다수의 세그먼트를 가지는 객체로 표현된다. 기존 연구의 질의 색인들은 영역 연속 질의 또는 간격 연속 질의와 같은 단순한 질의를 데이터로써 다루어왔다. 따라서, ECSpec에 기초한 연속 질의에 대하여 기존 질의 색인 기법을 적용할 경우 색인의 삽입 및 탐색 성능이 저하되는 문제가 있다. 즉, 하나의 연속 질의를 저장하기 위해서 다수의 세그먼트를 질의 색인에 삽입해야 하기 때문에 시간 및 공간 비용이 커진다. 반대로, 효과적인 삽입을 위해 세그먼트를 완전히 포함하는 최소 경계 사각형으로 처리할 경우에는 사장 영역이 매우 큰 데이터를 삽입하게 되므로, 탐색 시 다수의 데이터가 여과 및 정제 대상이 되어 탐색 성능이 저하되는 문제가 있다.

이 논문에서는 복잡한 RFID 연속 질의를 효과적으로 처리하기 위한 새로운 변환 기법을 제안한다. 이 기법은 연속 질의의 다수 세그먼트를 결집된 형태의 단일 데이터로 변환하고 이를 저장한다. 이를 위해서, RFID 연속 질의를 분석하여 세그먼트 간의 합동 관계와 규칙적 반복 특징이 존재함을 보인다. 이러한 특징에 기반하여 다수 세그먼트들의 반복되는 형태를 결정하고 이것을 새로운 변환 공간에 맵핑하여 단일 데이터로 표현하는 결집 변환(Aggregate Transformation) 기법을 제안한다. 또한, 이러한 변환 기법을 적용한 색인과 기존의 질의 색인의 성능 평가를 수행하여 그 결과를 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 RFID 미들웨어와 기존의 질의 색인의 관련 연구를 기술하고, 3장에서는 RFID 연속 질의를 색인 할 때의 문제점을 정의한다. 4장에서 연속 질의의 분석 결과 및 그 특징을 설명하고, 5장에서는 연속 질의를 단순화하기 위한 변환 기법을 제안하고 6장에서는 제안한 기법을 적용한 질의 색인과 기존 질의 색인들의 성능 비교를 수행하며 마지막으로 7장에서 결론을 맺는다.

2. 관련 연구

EPCglobal [1]은 RFID 애플리케이션을 위한 표준 질의 인터페이스인 ECSpec과 ECReportSpec을 제안하였다. 전자는 애플리케이션이 RFID 미들웨어로 질의를 보낼 때 사용되며 후자는 RFID 미들웨어가 질의 처리 결과를 애플리케이션에 전달하기 위해서 사용된다. ECSpec은 일정기간 동안 미들웨어에서 계속적으로 수행되는 질의이며 이러한 질의는 연속 질의(continuous query)라고 불린다. ECSpec은 다양한 질의를 지원하기 위한 인자인 predicates을 가지는데, 이 predicates는 판독기의 필터링 조건과 태그 필터링 조건으로 구성된다. 판독기 필터링 조건은 판독기 식별자 간격이며, 태그 필터링 조건은 다수의 태그 식별자 간격들을 나타내는 EPC Pattern으로 표현된다.

최근, RFID 스트리밍 데이터를 관리하기 위한 몇몇 연구들이 진행되었다. [15]는 RFID 데이터에서 나타나는 중복 데이터의 제거 및 노이즈 데이터의 제거를 위한 방법을 제안하였다. [5]에서는 RFID 데이터에 대한 시간기반 데이터 모델을 제안하고 이러한 데이터에 대하여 질의를 효과적으로 수행하기 위한 분할기반 데이터 관리 방법을 기술하였다. 그러나, 이러한 연구들은 RFID 연속 질의 처리의 성능 개선을 위한 질의 색인 기법에 대한 내용을 포함하지 않는다. 지금까지의 질의 색인 기법은 두 분야, 이동체 데이터에서의 질의 색인과 센서 데이터에서의 질의 색인에서 주로 연구되었다.

이동체 데이터베이스 환경에서는 이동체의 위치 스트리밍 데이터에 대한 영역 연속 질의를 처리하기 위한 질의 색인 기법이 연구되었다. CQI[3] 색인 기법은 영역 질의를 그리드 셀로 분할하고 분할에 사용된 셀의 list 구조에 질의의 식별자를 저장하는 방법이다. 즉, 이 list는 이동체가 위치를 보고할 때 해당 질의를 찾기 위한 대상이 된다. VCR[8] 색인 기법은 영역 질의를 가변 크기의 가상 구조(virtual constructs, VCs)를 이용하여 나누고 질의의 식별자는 각 가상 구조의 저장공간에 삽입된다. 따라서, 탐색은 이동체가 보고한 위치를 포함하는 가상 구조를 찾는 과정으로 구성된다. 이러한 기법들은 연속 질의를 사용자 모니터링 영역과 같은 단일 영역으로 고려하고 있다. 그러나, ECSpec에 기반한 RFID 연속 질의는 RID축의 간격과 TID축의 다수 간격으로 표현되기 때문에 색인 공간에서 다수의 세그먼트로 표현된다. 이러한 질의 색인 기법들을 RFID 질의 색인에 적용할 경우, 다수의 세그먼트를 삽입해야 하므로 삽입 시간이 증가하고 색인 크기도 증가하는 문제가 있다.

센서 네트워크 환경의 연속 질의를 처리하기 위해서, NiagaraCQ[6]과 TelegraphCQ[10]와 같은 센서 데이터 스트림 시스템은 IBS tree[4] 구조의 질의 색인을 활용하고 있다. IBS tree는 1차원의 균형 이진 트리이며, 질의 속성의 개수만큼 트리가 구축된다. 이 색인은 연속 질의에 명시되는 속성의 간격을 저장하고 센서 노드에서 감지 이벤트가 발생할 때 감지된 값을 만족하는 간격 데이터를 이진 탐색하여 결과셋을 도출한다. 그러나, 이 기법을 RFID 질의 색인으로 적용할 경우 각 속성값마다 구축된 트리를 탐색하고 그 결과를 합병해야 하는 문제가 있다.

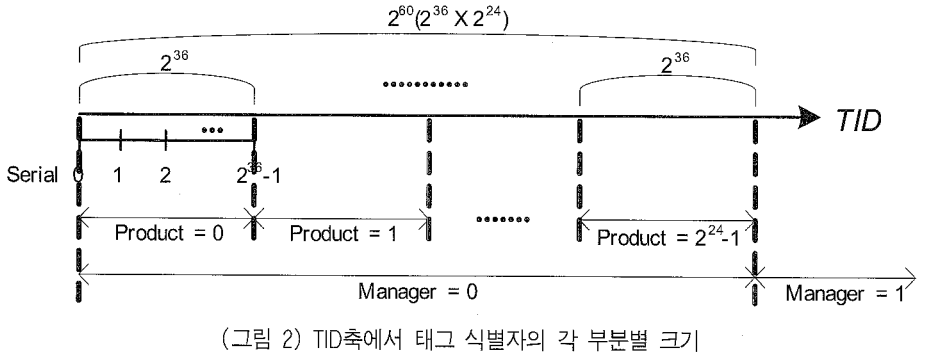
3. 문제 정의

ECSpec의 predicates은 판독기와 태그의 필터링 조건으로 구성된다. 판독기의 필터링 조건은 판독기 식별자 도메인(RID)에서 간격으로 표현된다. 반면에, 태그의 필터링 조건은 태그 식별자 도메인(TID)에서 다수의 간격을 나타내는 EPC Pattern으로 표현된다. 이 장에서는, EPC Pattern을 설명하고 이러한 predicates으로 구성되는 RFID 연속 질의의 문제점을 설명한다.

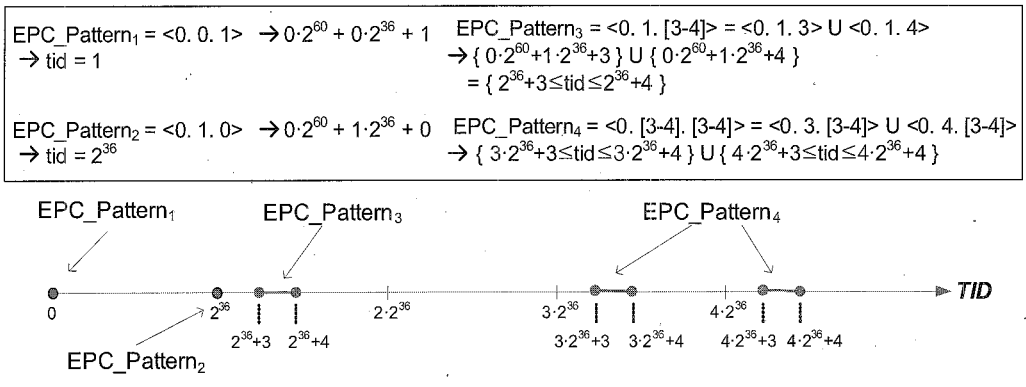
3.1 태그 식별자 도메인에서 EPC Pattern

EPCglobal의 TDS(Tag Data Standard)[2]는 64bits, 96bits, 128bits 등 태그의 여러 가지 포맷을 기술하고 있다. 이 논문에서는 가장 널리 사용되는 포맷인 GID-96(96bits)을 대상으로 설명한다. GID-96 태그는 8bits의 헤더와 88bits의 데이터로 구성되고 태그 식별자의 최대값은 2^{88}이다. 그리고 88bits의 데이터는 회사를 구분하는 Manager(최상위 28bits), 제품을 구별하는 Product(상위 24bits), 아이템을 구분하는 Serial(하위 36bits)로 구성된다. 따라서 그림 2에서와 같이 태그 식별자 도메인(TID)에서 Serial의 단위는 TID축의 1이고, Product의 단위는 TID축에서 2^{36}의 크기이며, Manager의 단위는 TID축에서 2^{60}의 크기이다.

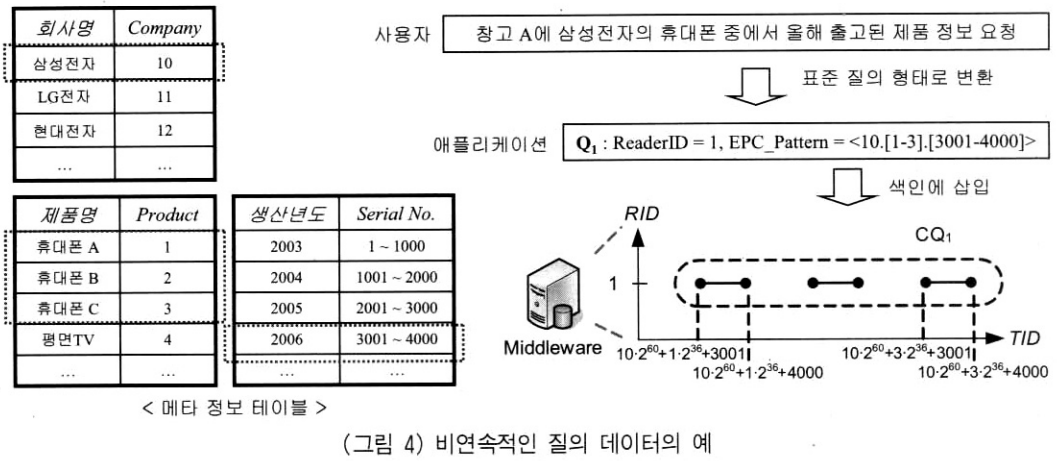
EPCglobal은 RFID 스트리밍 데이터의 수집 및 정제를 위한 표준 인터페이스인 ECSpec을 제시하였는데, 이것은



(그림 2) TID축에서 태그 식별자의 각 부분별 크기



(그림 3) EPC Pattern의 예



(그림 4) 비연속적인 질의 데이터의 예

판독기와 태그의 필터링 조건으로 구성된다. 판독기의 필터링 조건은 RID축의 간격으로 표현되지만, 태그 필터링 조건은 manager, product 그리고 serial 부분에 대해 각각 간격을 지정하는 방법인 EPC Pattern으로 표현된다. 예를들어, <A. B. C>의 EPC Pattern의 경우 A, B, C는 각각 manager, product, serial을 의미한다. [1]에 따르면, 이러한 EPC Pattern의 각 부분은 constant, [low-high] 또는 *로 표기되며, 각각 상수, 간격 그리고 임의값(또는 모든값)을 의미한다.

그림 3은 EPC Pattern의 예를 보여준다. *EPC_Pattern*$_1$ (<0.0.1>)은 TID축의 1이며, *EPC_Pattern*$_2$ (<0.1.0>)는 2^{36}을 의미한다. 그리고 *EPC_Pattern*$_3$ (<0.1.[3 - 4]>)은 {<0.1.3> ∩ <0.1.4>}를 의미하므로 {2^{36} + 3 ≤ tid ≤ 2^{36} + 4}의 간격을 의미한다. 그런데, *EPC_Pattern*$_4$ (<0.[3 - 4].[3 - 4]>)는 {<0.3.[3 - 4]> ∩ <0.4.[3 - 4]>}를 의미하므로 {3·2^{36} + 3 ≤ tid ≤ 3·2^{36} + 4}와 {4·2^{36} + 3 ≤ tid ≤ 4·2^{36} + 4}의 두 개의 간격을 가진다.

3.2 질의 색인에서 데이터의 문제점

RFID 스트리밍 데이터의 특징은 지속적으로, 빠르게, 끊임없이 그리고 실시간으로 생성된다는 것이다. 이러한 스트리밍 데이터에 대하여 전통적인 데이터 색인 기법을 사용할 경우, 태그가 인식될 때마다 색인의 갱신이 필요하다. 또한 RFID 스트리밍 데이터에 대한 질의는 일회성 질의(one-time query)가 아니라 일정기간 동안 수행되는 연속 질의(continuous query) 이다. 이러한 환경에서 연속 질의 처리의 성능을 향상시키기 위해서는 질의를 색인의 데이터로써 처리하는 질의 색인 기법이 적합하다[3, 8]. 그러나, RFID 질의 색인의 데이터 및 질의는 기존 연구에서 다루지 않았기 때문에 이를 정의하는 것이 필요하다.

Definition 1. *Stabbing 질의*는 판독기가 개별 태그를 인식할 때 발생하는 이벤트로써, 질의 색인의 데이터를 탐색하는 점질의이다. *StabbingQuery* = $\{(rid_i, tid_j) \in R^2\}$.

Definition 2. *질의 데이터*는 ECSpec으로부터 유도된 연속 질의이며, 질의 색인의 데이터가 된다. 이 데이터는 판독기와 태그의 필터링 조건으로 구성되는데, 판독기 조건은 RID축의 단일 간격 그리고 태그 조건은 TID축의 다수 간격으로 표현된다.

RFID 질의 데이터에 대하여, CQI[3] 또는 VCR[8]과 같은 2차원(RID, TID) 질의 색인이 기본적으로 RFID 질의 색인으로 적용될 수 있다. 이러한 2차원 공간에서 질의 데이터는 다수의 세그먼트로 구성되는 복합객체이다. 그림 4의 예제와 같이, 사용자가 "창고 A에 삼성전자 휴대폰 중 올해 출고된 제품" 정보를 원한다고 가정하고 창고 A에 식별자가 1인 판독기가 설치되었다고 가정하면 애플리케이션은 ECSpec Q1(readerID = 1, EPC_Pattern = <10. [1 - 3]. [3001 - 4000]>)을 RFID 미들웨어로 전달한다. Q1은 미들웨어에 도

착된 후 질의 색인에 질의 데이터로써 삽입되어야 하는데, 이 질의 데이터는 2차원(RID, TID) 공간에서 세 개의 세그먼트로 표현된다. 첫 번째 세그먼트는 {(1, $10\cdot2^{60}$ + $1\cdot2^{36}$ + 3001), (1, $10\cdot2^{60}$ + $1\cdot2^{36}$ + 4000)}이고 두 번째는 {(1, $10\cdot2^{60}$ + $2\cdot2^{36}$ + 3001), (1, $10\cdot2^{60}$ + $2\cdot2^{36}$ + 4000)}이며 세 번째는 {(1, $10\cdot2^{60}$ + $3\cdot2^{36}$ + 3001), (1, $10\cdot2^{60}$ + $3\cdot2^{36}$ + 4000)} 이다. 이것은 EPC Pattern의 각 부분이 영역으로 표현될 수 있기 때문이며 TDS[2]에 따르면 하나의 질의 데이터는 최대 2^{24}개의 세그먼트로 구성될 수 있다. 즉, 애플리케이션의 연속 질의를 저장하기 위해서 매우 많은 세그먼트를 삽입해야 하는 문제가 있다.

이러한 다수 삽입 문제를 피하기 위해서, 두 가지 방법이 적용될 수 있다. 첫째, 질의 데이터의 모든 세그먼트를 포함하는 최소 경계 사각형(MBR)을 2차원 질의 색인(CQI 또는 VCR)에 삽입하는 방법이 있다. 그러나 이 경우에 각 MBR은 매우 큰 사장 영역을 포함하게 되고, 이로 인해 탐색 시 정제(refinement) 단계가 필요하며 사장 영역에 stabbing 되는 질의를 마지막 정제 단계에서 제외시켜야 하므로(false hit) 탐색 성능이 저하되는 문제가 있다. 둘째, 색인의 공간을 2차원이 아닌 4차원(RID, Manager, Product, Serial)으로 구성하면, 질의 데이터를 사장 영역이 없는 단일 데이터로 삽입할 수 있다. 그러나 [14]에 따르면, 색인의 차원이 높아질수록 성능은 차원의 지수승으로 증가하는 문제가 발생하므로, 실시간 처리가 필요한 RFID 미들웨어를 위한 새로운 접근 방법이 필요하다. 본 논문에서는 다수의 세그먼트로 구성되는 질의 데이터를 3차원 공간에서 사장 영역이 없는 단일 데이터로 표현할 수 있는 변환 기법을 제안한다. 그리고 위에서 언급된 방법들에 대한 성능은 실험을 통해서 확인한다.

4. 질의 데이터의 분석

질의 색인의 데이터, 즉 질의 데이터는 EPC Pattern에 따라 형태가 결정되므로 질의 데이터의 특징을 도출하기 위해서는 EPC Pattern에 대한 case study를 수행하는 것이 필요하다. EPC Pattern의 각 부분은 상수, 간격 그리고 임의 값(또는 모든값)이므로 27가지의 경우를 분석하였다.

27가지 경우 중, 질의 데이터로써 존재 가능한 11가지의 의미있는 형태가 존재한다. EPC Pattern의 각 부분은 manager, product 그리고 serial이며 manager의 목적은 각 회사에 글로벌하며 유일한 식별자를 부여하기 위한 것이고 product와 serial은 각 회사내의 로컬 규칙에 따라 제품과 아이템의 유일한 식별자를 부여하기 위한 것이다. 따라서, <*[a1 - a2]*. *. *> and <*. *. *>를 제외하면 manager가 간격의 값을 가지는 질의 데이터는 존재하지 않는다. Case study의 결과에 따르면 질의 데이터는 단일 또는 복수의 세그먼트로 구성되며, 복수 개로 구성되는 질의 데이터의 경우 세그먼트의 수는 EPC Pattern의 product의 간격과 같다.

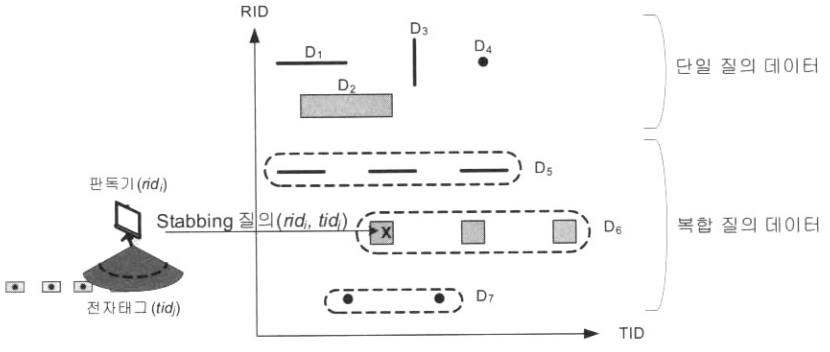
질의 데이터는 2차원 공간(RID, TID)에서 단일 또는 복수의 세그먼트로 구성된다. 단일 세그먼트로 구성되는 질의 데이터는 *단일 질의 데이터(simple query data)*라고 정의하고 2 ~ 2^{24}개의 세그먼트로 구성되는 질의 데이터를 *복합 질의 데이터(complex query data)*라고 정의하며 각 세그먼트를 *질의 데이터 세그먼트(segment of query data)*라고 정의한다. 질의 데이터 d 와 질의 데이터 세그먼트 d_i 와의 관계는 $d = \{d_1, \cdots, d_n\}$ $(1 \le n \le 2^{24})$로 표현된다. 질의 데이터 세그먼트는 2차원 공간의 각 모서리 값을 가지는 $d_i = \{(min_{rid}, min_{tid}), (max_{rid}, max_{tid})\}$ $(d_i \in d)$로 표현된다. 복합 질의 데이터를 구성하는 다수의 세그먼트들은 서로 기하학적인 관련성이 존재한다. 이 세그먼트들은 기하학적으로 서로 다른 위치에 존재하지만 크기와 모양이 동일한 관계, 즉 합동 관계를 가진다. 이 특징은 아래와 같이 정리 및 증명된다.

Theorem 1. 표 1에서 보여지는 것처럼, EPC Pattern의 일반 형식은 $<M_1.[P_1 - P_2].[S_1 - S_2]>$이다. 복합 질의 데이터는 RID축으로 $\overline{(rid_a, rid_b)}$의 간격을 가지며 TID축으로 EPC Pattern $<m_1.[p_1 - p_2].[s_1 - s_2]>$를 가진다고 가정하자. 그러면, 복합 질의 데이터 d는 질의 데이터 세그먼트 $\{d_0, d_1, \cdots, d_{p2-p1}\}$ 로 구성된다. 이 때, $0 \le i < j \le p_2 - p_1$인 i와 j에 대하여 d_i와 d_j는 서로 합동($d_i \equiv d_j$)이다.

Proof: i 가 0 일 경우, d_0 = {(rid_a, $m_1\cdot2^{60}$ + $p_1\cdot2^{36}$ + s_1), ((rid_b, $m_1\cdot2^{60}$ + $p_1\cdot2^{36}$ + s_2))} 이고 d_1 = {(rid_a, $m_1\cdot2^{60}$ + (p_1 + 1)$\cdot2^{36}$ + s_1), ((rid_b, $m_1\cdot2^{60}$ + (p_1 + 1)$\cdot2^{36}$ + s_2))} 이다. 이 식은 $d_1 = d_0 + (0, 2^{36})$ 로 정리된다. 즉, d_0 와 d_1 의 RID축 길이는 $\overline{(rid_a, rid_b)}$으로 같고, d_0와 d_1의 TID축 길이도 $s_2 - s_1$으로 동일하다. 따라서 $i = 0$에 대하여 $d_0 \equiv d_1$ 이다. $0 <$

〈표 1〉 EPC Pattern의 Case Study

a. b. c	[a1 - a2]. b. c	*. b. c
a. b. *	[a1 - a2]. b. *	*. b. *
a. b. [c1 - c2]	[a1 - a2]. b. [c1 - c2]	*. b. [c1 - c2]
a. *. c	[a1 - a2]. *. c	*. *. c
a. *. *	[a1 - a2]. *. *	*. *. *
a. *. [c1 - c2]	[a1 - a2]. *. [c1 - c2]	*. *. [c1 - c2]
a. [b1 - b2]. c	[a1 - a2]. [b1 - b2]. C	*. [b1 - b2]. c
a. [b1 - b2]. *	[a1 - a2]. [b1 - b2]. *	*. [b1 - b2]. *
a. [b1 - b2]. [c1 - c2]	[a1 - a2]. [b1 - b2]. [c1 - c2]	*. [b1 - b2]. [c1 - c2]



(그림 5) 질의 데이터와 Stabbing 질의

$n < p_2 - p_1$인 n에 대하여 $i = n$ 일 때 $d_n \equiv d_{n+1}$ 임을 가정하자. $i = n + 1$일 경우 $d_{n+1} = \{(rid_a, m_1 \cdot 2^{60} + (p_1 + n + 1) \cdot 2^{36} + s_1), (rid_b, m_1 \cdot 2^{60} + (p_1 + n + 1) \cdot 2^{36} + s_2))\}$이고 $d_{n+2} = \{(rid_a, m_1 \cdot 2^{60} + (p_1 + n + 2) \cdot 2^{36} + s_1), (rid_b, m_1 \cdot 2^{60} + (p_1 + n + 2) \cdot 2^{36} + s_2))\}$이다. 이 식을 정리하면, $d_{n+1} = d_n + (0, 2^{36})$이고 $d_{n+2} = d_{n+1} + (0, 2^{36})$ 이다. $d_n \equiv d_{n+1}$임을 가정하였으므로 $d_{n+1} \equiv d_{n+2}$가 성립된다. 따라서, $d_i \equiv d_{i+1}$이고, 결과적으로 $d_i \equiv d_j$ $(0 \le i < j \le p_2 - p_1)$ 관계가 성립한다.

복합 질의 데이터의 또 하나의 특징은 세그먼트들이 TID축에서 일정한 간격으로 배치된다는 것이다. 위 증명과정에서 나타나듯이 각 세그먼트는 동일한 RID축의 값을 가지면서 TID축으로 앞 세그먼트와 뒤 세그먼트간 일정한 거리(2^{36})를 두고 나타난다.

Lemma 1. 질의 데이터 세그먼트들은 TID축으로 일정한 거리를 두고 반복적으로 나타난다.

그림 5는 RID, TID의 2차원 공간상에 단일 질의 질의와 복합 질의 데이터의 형태를 보여준다. 특히, 복합 질의 데이터(D_5~D_7)의 세그먼트들은 합동 관계를 가지므로 동일한 모양으로 나타나고, TID축으로 일정한 간격(2^{36})으로 존재함을 보여준다. 그리고 Stabbing 질의는 판독기에서 태그를 인식하면서 발생하고 질의 색인에서 점질의로써 처리됨을 보여준다.

5. 결집 변환

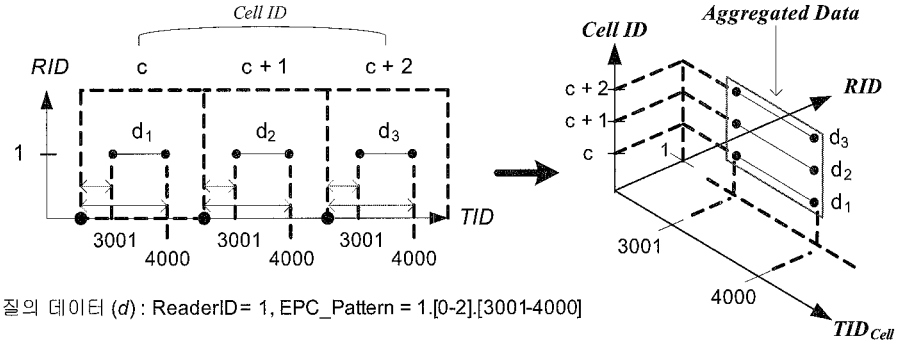
복합 질의 데이터를 단순화하기 위해서는 데이터의 표현법을 변환(Transformation)하는 기법이 필요하다. 기존 색인 연구에서의 변환의 개념은 크게 두 가지 방법으로 제안되었다[14]. 첫 번째는 저차원의 객체를 고차원의 점(Point)으로 표현하는 방법으로써 끝점 변환(endpoint Transformation), 중점 변환(midpoint Transformation)이 있다. 두 번째는 공간을 매우 작은 그리드 셀들로 나누고 순서화 한 후, 2차원의 객체를 1차원의 간격으로 표현하는 방법으로써 Z-Ordering, Hilbert R tree와 같은 연구가 진행되었다. 기존 방법과 달리, 이 논문에서는 다수의 세그먼트들로 구성되는 복합 질의 데이터를 단순화된 객체로 표현하기 위한 변환 방법을 제시한다.

5.1 기본 아이디어(Basic Idea)

복합 질의 데이터의 모든 세그먼트를 질의 색인에 삽입하는 것은 삽입 시간 및 저장 공간 비용이 증가하는 문제가 있기 때문에 적합하지 않다. 이 논문에서는 복합 질의 데이터를 단순화된 형태로 표현하기 위해서, Theorem 1과 Lemma 1에서 기술된 질의 데이터 세그먼트간 특징을 이용하는 결집 변환(Aggregate Transformation) 기법을 제안한다. 복합 질의 데이터의 세그먼트들은 모양 및 크기가 동일한 합동 관계 특성과 규칙적으로 반복되는 특성을 가진다. 이러한 데이터를 단순화하기 위해서는 규칙적으로 반복되는 형태를 찾는 것이 중요한데 이 때, 그리드 구조는 일정한 크기의 셀들이 반복적으로 배치되므로 복합 질의 데이터로부터 규칙적으로 반복되는 형태를 추출하기 위한 도구로 유용하게 사용될 수 있다.

이 논문의 기본 아이디어는 그림 6에서 보여진다. 그림 6-(가)는 복합 질의 데이터 d가 2차원 공간에서 세 개의 이산적인 세그먼트 d_1, d_2와 d_3로 구성됨을 보여준다. 이 때, 복합 질의 데이터가 TID축으로 2^{36}의 크기를 가지고 RID축으로 최대값을 가지는 셀들로 구성된 고정 그리드 구조와 오버레이된다고 가정하자. 그러면 Theorem 1과 Lemma 1에 의해 그림 6-(가)에서 보여지는 것처럼 d_1, d_2 와 d_3 는 각 세그먼트를 포함하는 셀의 원점으로부터 동일한 모양이 되며 각 세그먼트는 서로 다른 셀 식별자(c, c+1, c+2)를 가진다. 만약, 단일 셀 공간(RID, TID_{cell})에 셀 식별자 차원(Cell ID)을 추가한 3차원 공간에서 d_1, d_2와 d_3를 나타내면 그림 6-(나)에서 보여지는 것처럼 셀 공간의 세그먼트들이 셀 식별자 차원에서 순차적으로 배치됨을 알 수 있다. 따라서, 이러한 세그먼트 d_1, d_2 와 d_3 가 셀 식별자 차원에서 연속적이므로 이를 모으면 단일 데이터로 표현할 수 있다. 복합 질의 데이터의 모든 세그먼트를 대표하는 이 객체를 결집 데이터(aggregated data)라고 하며, {(1, 3001, c), (1, 4000, c+2)}같이 나타낸다.

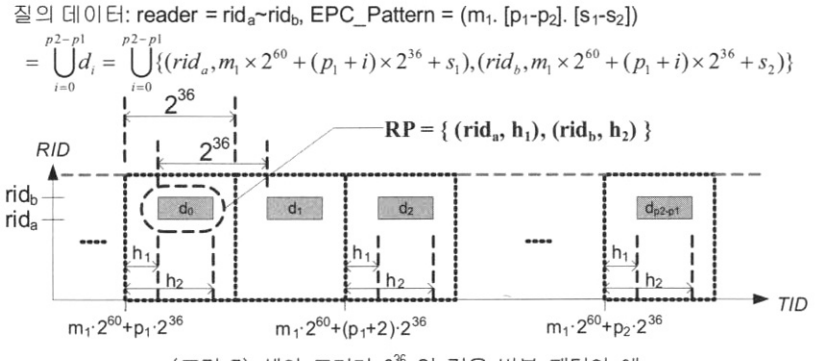
결집 변환은 복합 질의 데이터의 다수 세그먼트를 저장할 때 발생하는 색인상의 삽입 시간 및 저장 공간 문제를 해결한다. 이 기법의 핵심 아이디어는 복합 질의 데이터 대신, 복합 질의 데이터로부터 규칙적으로 반복되는 형태를 추출한 결집 데이터로 표현하는 것이다. 이것이 이 논문에서 제안하는 변환에 의한 단순화 방법이다.



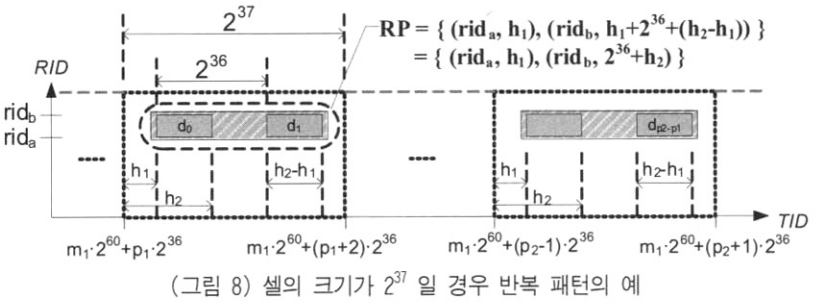
복합 질의 데이터 (*d*) : ReaderID= 1, EPC_Pattern = 1.[0-2].[3001-4000]

(가) 2D에서의 복합 질의 데이터 (나) 3D에서의 결집 데이터

(그림 6) 결집 변환의 예



(그림 7) 셀의 크기가 2^{36} 일 경우 반복 패턴의 예



(그림 8) 셀의 크기가 2^{37} 일 경우 반복 패턴의 예

5.2 결집 변환의 과정

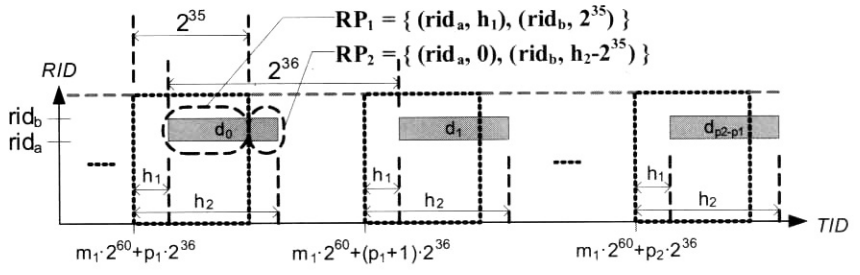
결집 변환 과정은 세 단계로 구성된다. 첫째, 그리드 구조를 이용하여 질의 데이터로부터 반복되는 형태를 추출한다. 둘째, 그리드 구조에서 질의 데이터를 포함하는 셀들의 최소 경계 식별자 간격을 구한다. 마지막으로, 추출한 반복 형태와 식별자 간격을 결집하여 변환된 공간에서의 단일 데이터로 변환한다. 이러한 과정을 관련 용어를 정의하면서 기술한다.

Definition 3. *반복 패턴(Repeated Pattern, RP)*은 2차원 공간에서 질의 데이터가 그리드 셀과 오버레이될 때 셀 내부에서 질의 데이터의 형태를 의미하며 $RP = \{((min_{rid}, min_{tid}), (max_{rid}, max_{tid})) \subset R^2\}$ 로 표현된다.

변환의 첫 단계는 질의 데이터로부터 반복 패턴을 추출하는 것이다. 질의 데이터는 RID축으로 $\overline{(rid_a, rid_b)}$의 간격과 TID축으로 EPC Pattern $<m_1.[p_1 - p_2].[s_1 - s_2]>$을 가진다고 가정하자. $d_0, d_1, \cdots, d_{p2-p1}$를 질의 데이터의 세그먼트라고 하고 d_i의 좌하단 점을 포함하는 셀의 좌하단 점에서 세그먼트 d_i까지의 TID축의 최소 및 최대 거리를 각각 h_1와 h_2라고 하자.

반복 패턴을 추출하는 방법은 그리드 셀의 크기에 따라서 두 가지 경우로 나뉜다. 그림 7은 그리드 셀의 TID축 크기가 질의 데이터의 세그먼트간 거리(2^{36})보다 같거나 작을 경우를 보여주는데, 여기서는 셀의 크기를 2^{36}이라고 가정하자. 이미 살펴본 것처럼, 세그먼트들은 TID축으로 2^{36} 마다 반복된다. 이 때, Theorem 1과 Lemma 1에 따르면 각 세그먼트를 포함하는 셀의 공간에서 세그먼트 $d_0, \cdots, d_{p2-p1}$는 동일한 형태가 되는데, 이 형태가 반복 패턴이다. 정의에 따라서, 그림 7의 반복 패턴은 $\{(rid_a, h_1), (rid_b, h_2)\}$로 표현된다.

그림 8은 그리드 셀의 TID축 크기가 질의 데이터의 세그먼트간 거리(2^{36})보다 클 경우를 보여주는데, 여기서는 셀의

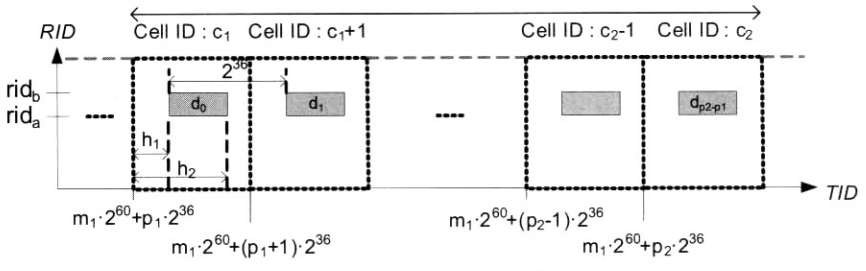


(그림 9) 반복 패턴 분할의 예

질의 데이터: reader = rid_a~rid_b, EPC_Pattern = (m_1. [p_1-p_2]. [s_1-s_2])

$$= \bigcup_{i=0}^{p2-p1} d_i = \bigcup_{i=0}^{p2-p1} \{(rid_a, m_1 \times 2^{60} + (p_1 + i) \times 2^{36} + s_1), (rid_b, m_1 \times 2^{60} + (p_1 + i) \times 2^{36} + s_2)\}$$

$$RC = \overline{(FindCell(rid_a, d_0.\min_{tid}),\ FindCell(rid_b, d_{P2-P1}.\max_{tid}))} = \overline{(c_1,\ c_2)}$$



(그림 10) 셀 간격의 예

크기를 2^{37}이라고 가정하자. 이 경우, 셀 내부에 두 개의 세그먼트가 존재하게 되고, 두 세그먼트를 포함하는 그룹이 규칙적으로 반복해서 나타난다. 따라서 이 그룹이 반복 패턴이 되고 $\{(rid_a, h_1), (rid_b, 2^{36} + h_2)\}$와 같이 표현된다. 일반적으로 그리드 셀의 TID축 크기(2^K)가 세그먼트간 거리(2^{36}) 보다 클 경우, 하나의 셀에 포함되는 세그먼트의 수는 2^{K-36} 이다.

그리드 크기에 따른 두 가지 경우와 상관없이, 질의 데이터의 개별 세그먼트가 둘 이상의 셀과 겹쳐질 경우에는 그리드 셀의 경계선에 의해 각 세그먼트는 분리된다. 그 이유는 반복 패턴은 셀 내부에서 나타나는 질의 데이터의 모양을 의미하기 때문이다. 그림 9는 이러한 분할 예제를 보여준다. 그리드 셀의 TID축 크기를 2^{35}이라고 가정하고 최소 및 최대 거리의 관계를 $h_1 < 2^{35} < h_2$ 라고 가정하자. 이 경우, 모든 세그먼트는 그림 9와 같이 두 개의 셀과 겹치게 된다. 반복 패턴의 정의에 따라서 두 개의 반복 패턴이 추출되는데, 첫 번째 반복 패턴(RP_1)은 $\{(rid_a, h_1), (rid_b, 2^{35})\}$가 되고 두 번째 반복 패턴($RP_2$)는 $\{(rid_a, 0), (rid_b, h_2 - 2^{35})\}$가 된다. 세그먼트의 크기는 TID축으로 2^{36}보다 항상 작기 때문에 그리드 셀의 TID축 크기가 2^{36}보다 같거나 클 경우에는 이러한 분할이 발생하지 않는다.

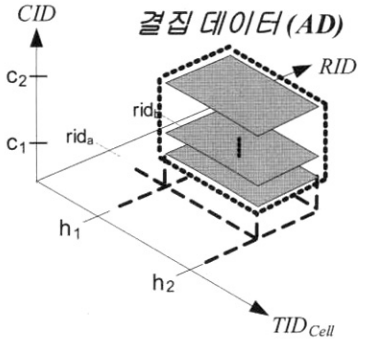
Definition 4. *셀 간격(Range of Cells, RC)*은 2차원 공간에서 질의 데이터가 그리드 셀과 오버레이될 때 질의 데이터를 포함하는 셀들의 최소 식별자 간격을 의미하며 $RC = \{\overline{(\min_{cid},\ \max_{cid})} \subset R\}$ 로 표현된다.

변환의 두 번째 단계는 질의 데이터로부터 셀 간격을 결정하는 것으로써, 셀 간격은 질의 데이터를 완전히 포함하는 셀들의 최소 식별자와 최대 식별자로 구성된다. 그림 10은 그림 7에서 기술된 질의 데이터에 대한 셀 간격을 보여준다. 그림에서 보여지는 것처럼, 셀 간격의 최소값은 세그먼트 d_0를 포함하는 셀의 식별자이고 최대값은 세그먼트 d_{P2-P1}를 포함하는 세그먼트의 식별자이다. 이러한 셀의 식별자는 (rid_a, $d_0.min_{tid}$)와 (rid_b, $d_{P2-P1}.max_{tid}$)의 각 점을 셀 식별자를 찾는 그리드의 해쉬 함수에 인자로 사용하여 계산된다. 따라서, 그림 10에서 질의 데이터의 셀 간격은 $\overline{(c_1, c_2)}$가 된다.

Definition 5. *결집 데이터(Aggregated Data, AD)*는 질의 데이터의 모든 세그먼트를 대표하는 데이터로써, 반복 패턴과 셀 간격을 변환 공간에서 결집(aggregate)하여 생성되는 단일 데이터이다. 결집 데이터는 $AD = \{((min_{rid}, min_{tid}, min_{cid}), (max_{rid}, max_{tid}, max_{cid})) \subset R^3\}$ 로 표현된다.

변환의 마지막 단계는 변환 공간에 반복 패턴과 셀 간격을 이용하여 결집 데이터를 생성하는 것이다. 우선, 변환 공간은 RID, TID_{cell}, CID축으로 구성된다. RID와 TID_{cell}은 단일 그리드 셀의 공간이고 CID는 그리드의 개별 셀을 구분하기 위한 식별자 차원을 의미한다. 이러한 변환 공간에서 결집 데이터는 2차원 공간(RID, TID_{cell})의 반복 패턴을 CID축의 셀 간격만큼 결집한 것이다. 그림 11은 결집 데이터의 한 예를 보여준다. 질의 데이터와 그리드 셀의 크기가 그림 7에서 기술된 형태라고 가정하자. 이 경우 그림 7과 10에서

$$RP = \{(rid_a, h_1), (rid_b, h_2)\}$$
$$RC = \overline{(c_1,\ c_2)}$$
$$\Rightarrow\ AD = \{\ (rid_a, h_1, c_1),\ (rid_b, h_2, c_2)\ \}$$



(그림 11) 결집 데이터의 예

〈표 2〉 복합 질의 데이터의 변환식

Cellsize	세그먼트 크기	겹침 셀 수	결집 데이터
$\leq 2^{36}$	$< Cell_{size}$	1	AD = {(rid_a, h_1, c_1), (rid_b, h_2, c_2)}
		≥ 2	AD1 = {(rid_a, h_1, c_1), (rid_b, $Cell_{size}$, c_2)} AD_2 = {(rid_a, 0, c_1+1), (rid_b, h_2 MOD $Cell_{size}$, c_2+1)}
	≥ $Cell_{size}$	1	None
		≥ 2	AD = {(rid_a, h_1, c_1), (rid_b, h_2, c_2)}
$> 2^{36}$	$< Cell_{size}$	1	AD = {(rid_a, $h_1-(p_1$ MOD $2^{K-36})\cdot 2^{36}$, c_1), (rid_b, $h_2+(p_1$ MOD $2^{K-36})\cdot 2^{36}$, c_2)}
		≥ 2	None
	≥ $Cell_{size}$	1	None
		≥ 2	None

설명된 것처럼, 반복 패턴은 {(rid_a, h_1), (rid_b, h_2)}이고 셀 간격은 $\overline{(c_1, c_2)}$이다. 이 두 데이터를 변환 공간에 나타내면, 그림 11에서 보여지는 것처럼 2차원 공간(RID, TID_{cell})의 반복 패턴이 CID축에서 c_1 에서 c_2 까지 점유하게 되는데, 이것은 다음과 같이 표현된다.

$$\bigcup_{i=c_1}^{c_2} \{(rid_a, h_1, i), (rid_b, h_2, i)\} \quad (1)$$

수식 1의 모든 구성 요소들이 CID축에 순차적으로 배치되어 있기 때문에 전체 점유된 공간을 포함하는 새로운 객체를 생성할 수 있는데, 이것이 그림 11에서 보여지는 결집 데이터이다. 결집 데이터는 모서리 표현법(corner representation)에 의해 {(rid_a, h_1, c_1), (rid_b, h_2, c_2)}로 표현된다.

결집 변환은 복합 질의 데이터를 3차원 공간에서 단일 데이터로써 표현하는 변환 기법이다. 이 기법의 장점은 질의 데이터의 다수 세그먼트를 삽입하기 위한 삽입 시간 및 저장 공간의 문제를 해결할 수 있다는 것이다. 결집 변환은 세가지 요소에 영향을 받는데, 그 요소는 그리드 셀의 크기, 질의 데이터 세그먼트의 크기 그리고 세그먼트와 셀의 겹침이다. 지금까지 살펴본 내용을 이러한 요소에 따라 분류하면 변환식을 도출할 수 있다. 질의 데이터가 RID축으로 $\overline{(rid_a, rid_b)}$의 간격 그리고 TID축은 EPC pattern <m_1.[p_1 - p_2].[s_1 - s_2]>로 구성됨을 가정하자. 이러한 질의 데이터의 세그먼트를 각각 d_0, d_1, … ,d_{p2-p1}라고 하고 그리드 셀의 TID축 크기를 $Cell_{size}$ 라고 하자. 또한, (rid_a, $d_0.min_{tid}$)를 포함하는 셀의 식별자를 c_1, (rid_b, $d_{b2-b1}.max_{tid}$)를 포함하는 셀의 식별자를 c_2라고 하고 세그먼트 d_i와 d_i의 좌하단 점을 포함하는 셀의 좌하단 점까지 최소 및 최대 거리를 h_1와 h_2라고 하자. 이 때의 변환식은 표 2와 같다.

6. 성능평가

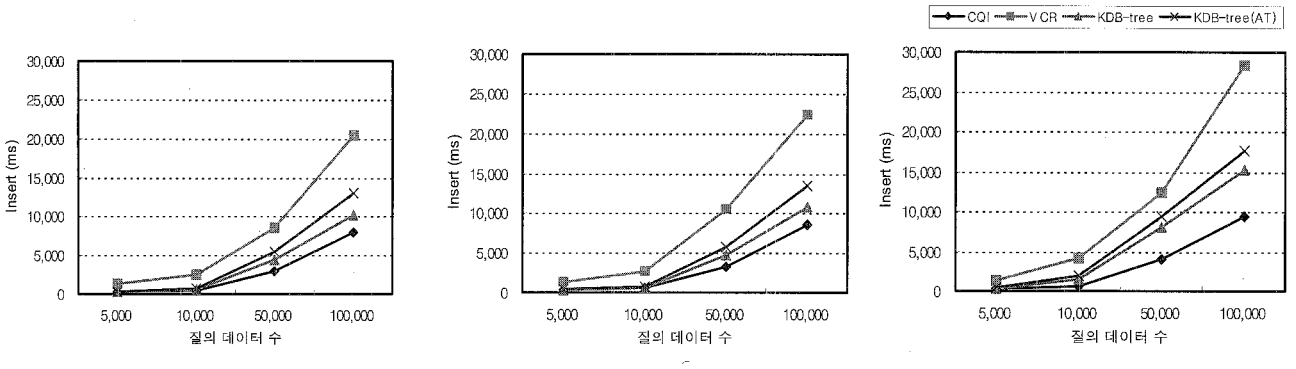
결집 변환의 효과를 파악하기 위해서 결집 변환 기법을 적용할 색인을 선정해야 한다. KDB-tree[7]는 동적 삽입에 대하여 트리의 균형을 유지하고 공간 기반의 색인을 구성하기 때문에 점질의에 대해서 다차원 색인들 중에서 성능이 우수하다[14]. RFID 질의 색인에서는 Stabbing 질의가 유일하며, 변환된 공간에서 점질의로 수행되기 때문에 KDB-tree가 변환 공간의 질의 색인으로써 적합할 것이다. 이 논문에서는 KDB-tree에 변환 기법을 적용한 KDB-tree(AT)를 구현하여 기존 기법들과 성능 비교를 수행하였다.

KDB-tree(AT)는 KDB-tree의 탐색(Stabbing 질의) 알고리즘, 삽입 알고리즘을 확장하였다. Stabbing 질의 (rid_i, tid_j)는 판독기(rid_i)가 태그(tid_j)를 개별적으로 인식할 때 발생하는 것으로 정의 1에서 기술된 것처럼 질의 색인에 저장된 데이터 중에서 이 점을 포함하는 질의 데이터를 찾는 것으로, Stabbing 질의는 변환 공간에서의 질의로 변환되어야 한다.

```
StabbingQuery(root, q)
S1 : if q is not transformed yet
     cid_k = FindCell(q)
     tid_cell-j = EvaluateDist(cid_k, q)
     q = (q.rid, tid_cell-j, cid_k)
S2 : set N to be the root
     if N is a leaf
          for each entry e
          if Contain(e, q) is TRUE then add e to list of results
          else
               for each entry e
               if Contain(e, q) is TRUE then StabbingQuery(e, q)
     endif
     return list of results
```

```
Insert(root, obj)
S1 : if obj is not transformed yet
     RP = ExtractPattern(obj)
     RC = CalculateCellRange(obj)
     Obj = AggregateTransformation(RP, RC)
     endif
S2 : set N to be the root
     if N is a leaf
          if # of entries > fanout of leaf node
               insert obj in the node
          else
               Split(N, N_new, obj)
     else
          for each entry e
               if Overlap(e, obj) is TRUE then Insert(N, obj)
     endif
```

(그림 12) 탐색 및 삽입 알고리즘



(가) 균등 분포 (나) 가우시안 분포 (다) 편향 분포

(그림 13) 삽입 실험 결과

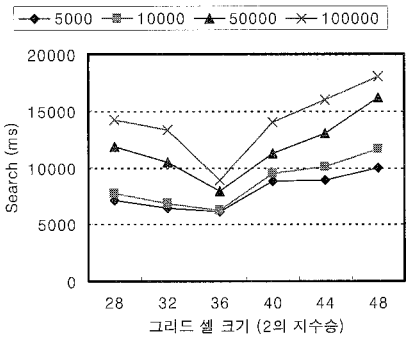
반복 패턴과 셀 간격 추출을 위한 그리드 구조에서 Stabbing 점(rid_i, tid_j)을 포함하는 셀의 식별자를 구하여 이를 cid_k 라고 하고, 셀 cid_k 내에서 이 Stabbing 점의 상대적 위치를 구하여 이를 tid_{cell-j}라고 하자. 이러한 과정을 거쳐, Stabbing 질의 (rid_i, tid_j)는 변환 공간에서의 점질의 (rid_i, tid_{cell-j}, cid_k)로 변환된다. 질의 데이터의 삽입 알고리즘은 질의 데이터의 변환과 변환된 데이터의 삽입 과정으로 구성된다. 질의 데이터는 5.2절에서 설명된 변환 과정에 의해 3차원의 결집 데이터로 변환되고, 변환된 질의 데이터는 질의 색인에 삽입된다. 이때 삽입되는 데이터와 비단말 노드의 엔트리들과 겹침이 발생할 경우, 겹침이 발생한 모든 엔트리에 데이터를 저장하는 방식을 선택한다. 이러한 중복 저장 방식은 Stabbing 질의의 수행 시, 단일 패스 검색을 보장하기 때문에 기존 KDB-tree의 단일 저장 정책 보다 질의 성능을 향상시킬 수 있다.

실험은 다양한 데이터 셋에 대하여 기존 질의 색인인 CQI[3], VCR[8], 오리지널 KDB-tree[7] 및 KDB-tree(AT)의 색인의 성능을 비교하였다. 실험 환경은 인텔 펜티움 IV 2.6 GHz 프로세서, 1 GB 메모리 그리고 마이크로소프트 윈도우 2003 서버 운영체제를 사용하는 개인용 컴퓨터에서 수행되었다. 실험에 사용된 색인은 모두 메모리에 상주하도록 하였으며, CQI와 VCR 그리고 KDB-tree 색인의 경우, 다수의 세그먼트를 삽입했을 경우 삽입 시간이 매우 오래 걸리기 때문에, 세그먼트들을 바운딩하는 최소 경계 사각형을 색인에 삽입하고 탐색 시 정제 단계를 수행하는 방식으로 구현되었다. 현재까지 알려진 RFID 데이터 셋이 존재하지 않기 때문에, 이 논문에서는 차세대물류IT기술연구사업단의 RFID 테스트 센터[1]로부터 dataset을 지원받아 실험을 수행하였다. 실험 데이터 셋은 균등 분포, 가우시안 분포 그리고 편향 분포로 구성되며 탐색 성능을 측정하기 위해서 100,000번의 stabbing 질의를 수행하였다.

그림 13은 데이터 셋에 따른 각 색인의 삽입 성능을 보여준다. 데이터 셋은 각 분포마다 5000, 10,000, 50,000 그리고 100,000개의 질의 데이터로 구성된다. CQI는 삽입 대상 셀에 데이터를 저장하는 방식이므로 삽입 연산이 단순하여 전체적으로 삽입 성능이 우수하였으며, 데이터 분포에 따른 변화도 적었다. VCR는 다양한 가상구조를 이용하여 데이터를 분할하여 삽입하는 연산 비용 때문에 성능이 낮았다. KDB-tree와 KDB-tree(AT)는 유사한 성능을 보였으나, 변환 연산이 추가된 KDB-tree(AT)가 삽입 시간이 증가하였다.

그림 14는 KDB-tree에 결집 변환을 적용할 때 효과적인

1) 차세대 물류 정보 기술을 개발하고 유비쿼터스 및 RFID 시스템과 같은 물류 IT 기반기술을 연구 개발하기 위해서 설립된 부산대학교 소재의 RFID 테스트 센터.

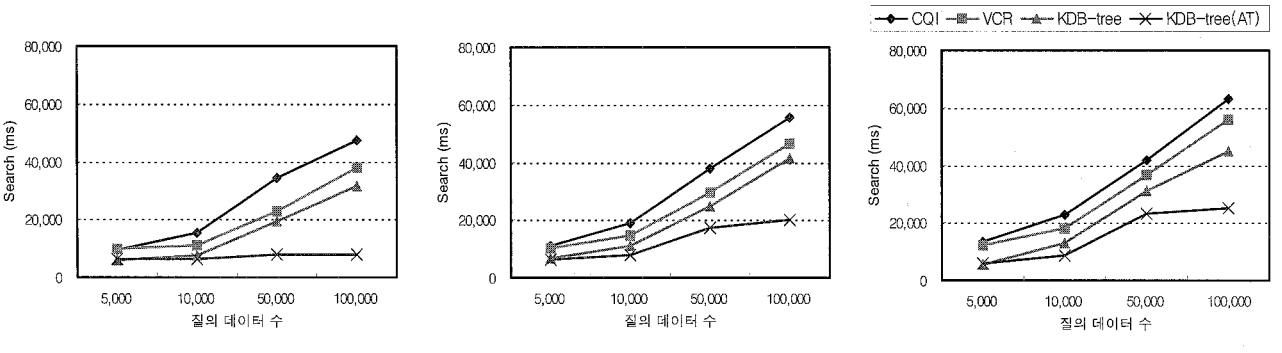


(그림 14) 셀 크기별 실험 결과

맵핑 그리드 셀의 크기를 실험한 결과를 보여준다. 데이터 셋은 5000, 10,000, 50,000 그리고 100,000개의 질의 데이터로 구성되는 균등 분포에 대하여 100,000번의 stabbing 질의를 수행했을 때의 탐색 시간을 측정하였다. 실험 결과는 그리드 셀의 TID축 크기가 2^{36}일 때 가장 성능이 좋으며 작거나 커질 경우 탐색 시간이 증가하였다. 그 이유는 셀의 크기가 2^{36}보다 작거나 클 경우에는 변환 공간에서 결집 객체가 빈 공간을 포함하게 되므로 정제 연산을 필요로 하기 때문이다. 이 후, 실험에서 KDB-tree(AT)의 셀 크기는 2^{36}으로 설정하여 수행하였다.

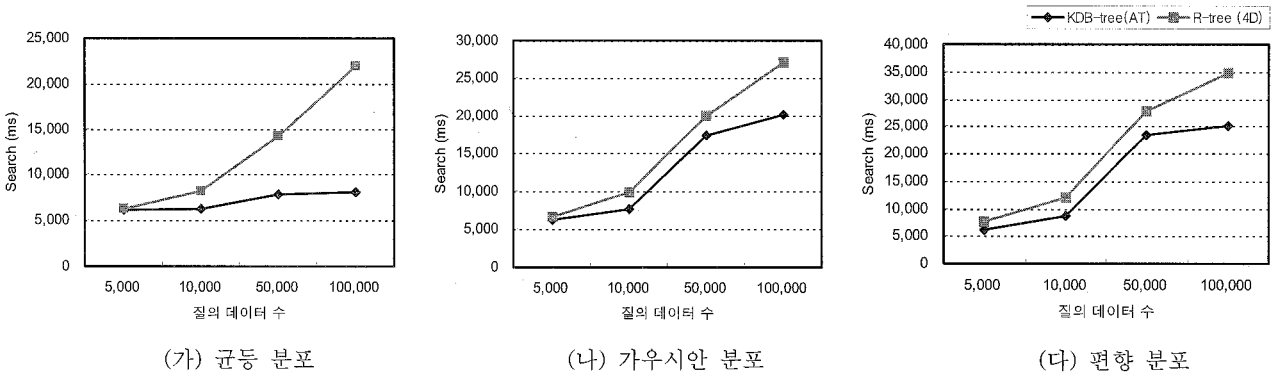
그림 15의 실험 결과는 다양한 데이터 셋에 대한 탐색 성능을 보여준다. 전체적으로, KDB-tree(AT)가 다른 색인보다 성능이 우수하였다. 질의 데이터 개수가 적을 때는 성능이 유사하였으나 질의 데이터 수가 증가할수록 CQI와 VCR의 성능 저하 폭이 커지는 현상이 나타났다. 그 이유는 변환 기법이 적용된 색인과 달리 이들 기존 색인에서는 탐색 시 데이터 별 정제 단계의 연산이 필요하고 이 연산의 수행 횟수는 데이터가 증가할수록 급격히 증가하기 때문이다. 또한, KDB-tree보다 변환 기법을 적용한 KDB-tree(AT)의 성능이 향상되었음을 알 수 있다.

그림 16은 대표적인 다차원 공간 색인인 R-tree와 KDB-tree(AT)의 탐색 성능을 보여준다. KDB-tree(AT)는 제안된 변환 기법을 통한 단일화가 가능하기 때문에 3차원 공간(RID, TID_{cell}, CID)으로 구현된 반면, 기존 R-tree는 다수의 세그먼트로 구성되는 질의 데이터를 단일 데이터로 삽입하기 위해서 4차원 공간(RID, manager, product, serial)으로 구현되었다. 실험 결과, 질의 데이터가 적을수록, 그리고 질의 데이터의 분포가 편향분포에 가까울수록 성능의 차이가 적었다. 하지만 균등 분포일수록 그리고 질의 데이터 수가 증가할수록 적은 차원에서 색인을 구축한 KDB-tree(AT)의 성능이 우수한 것으로 나타났다. 즉, 기존의 질의 색인 또는 기존 다차원 색인을 RFID 질의 색인으로 적용하는 것보다 본 논문에서 제안한 변환 기법을 KDB-tree에 적용한 질의 색인이 우수한 성능을 보였다.



(가) 균등 분포 (나) 가우시안 분포 (다) 편향 분포

(그림 15) 탐색 실험 결과 (2차원 색인과의 비교)



(가) 균등 분포 (나) 가우시안 분포 (다) 편향 분포

(그림 16) 탐색 실험 결과 (4차원 색인과의 비교)

7. 결 론

질의 색인 기법은 일반적으로 스트리밍 데이터에 대한 연속 질의 처리 환경에 적합하다. 기존 연구의 질의 색인은 영역 연속 질의 또는 간격 연속 질의와 같은 단순한 연속 질의만을 데이터로써 고려하였다. 그러나, 표준 질의 인터페이스인 ECSpec에 기초한 RFID 질의 색인의 데이터는 다수의 세그먼트로 나타나는 특징이 있다. 이러한 RFID 질의 색인으로써 기존 질의 색인을 이용할 경우, 하나의 연속 질의를 저장하기 위해서 다수의 세그먼트를 질의 색인에 삽입해야 하는 문제가 있다. 뿐만 아니라, 효과적인 삽입을 위해 세그먼트를 완전히 포함하는 최소 경계 사각형으로 처리할 경우에는 사장 영역이 매우 큰 데이터를 삽입하게 되므로, 탐색 시 다수의 데이터가 여과 및 정제 대상이 되어 탐색 성능이 저하되는 문제가 있다.

이 논문에서는 복잡한 RFID 연속 질의를 효과적으로 처리하기 위한 변환 기법을 제안하였다. 먼저, 연속 질의가 가지는 다수의 세그먼트간의 합동 관계성과 규칙적인 반복 특성을 발견하고 이를 증명하였다. 이러한 특징을 기반으로, 다수의 세그먼트에 대하여 반복되는 형태을 결정하고 이것을 단일 데이터로 변환하는 기법인 결집 변환을 제안하였다. 즉, 이 기법은 다수의 세그먼트를 사장영역 없이 단일 데이터로 색인에 저장할 수 있도록 한다. 그리고 다양한 데이터 셋에 대하여 기존의 질의 색인들과 이 변환 기법을 적용한 KDB-tree의 성능 비교를 수행하여 그 결과를 분석하였다.

참 고 문 헌

[1] EPCglobal, "The Application Level Event (ALE) Specification, Version 1.0," EPCglobal Standard Specification, 2005.

[2] EPCglobal, "EPCTM Tag Data Standards Version 1.3," EPCglobal Standard Specification, 2005.

[3] D. V. Kalashnikov, S. Prabhakar, W. G. Aref and S. E. Hambrusch, "Efficient evaluation of continuous range queries on moving objects," In Proc. 13th DEXA, pp.731 - 740, 2002.

[4] E. N. Hanson et al, "A Predicate Matching Algorithm for Database Rule Systems," ACM SIGMOD, pp.271 - 280, 1990.

[5] F. Wang and P. Liu, "Temporal Management of RFID Data," In Proc. 31st VLDB Conf., pp.1128 - 1139, 2005.

[6] J. Chen et al, "NiagaraCQ: A Scalable Continuous Query System for Internet Databases," ACM SIGMOD, pp.379 - 390, 2000.

[7] J. T. Robinson, "The K-D-B-tree: A search structure for large multidimensional dynamic indexes," ACM SIGMOD, pp.10 - 18, 1981.

[8] K L. Wu, S-K. Chen and P. S. Yu, "Processing Continual Range Queries over Moving Objects Using VCR Based Query Indexes," MobiQuitous 2004, pp.226 - 235, 2004.

[9] L. Golab and M. T. Ozsu, "Issues in Data Stream Management," ACM SIGMOD Record, pp.5 - 14, 2003.

[10] S. Chandrasekaran et al, "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World," In Proc. First Biennial Conf. on Innovative Data Systems Research, pp.269 - 280, 2003.

[11] S. R. Madden, M. A. Shah, J. M. Hellerstein and V. Raman, "Continuously adaptive continuous queries over streams," ACM SIGMOD, pp.49 - 60, 2002.

[12] S. S. Chawathe, V. Krishnamurthy, S. Ramachandrany and S. Sarma, "Managing RFID Data," VLDB, pp.1189 - 1195, 2004.

[13] S. Sarma, "Integrating RFID," ACM Queue, pp.50 - 57, 2004.

[14] V. Gaede and O. Günter, "Multidimensional Access Methods," ACM Computing Surveys, pp.170 - 231, 1998.

[15] Y. Bai, F. Wang and P. Liu, "Efficiently Filtering RFID Data Streams," Proc. of the First Int'l VLDB Workshop on Clean Databases(CleanDB'06), 2006.

박 재 관

e-mail : jkpack@pusan.ac.kr

1999년 부산대학교 컴퓨터공학과 (학사)

2001년 부산대학교 대학원 컴퓨터공학과 (공학석사)

2003년 부산대학교 대학원 컴퓨터공학과 (박사과정 수료)

관심분야 : 유비쿼터스 시스템, RFID 미들웨어, 스트림데이터 처리기

홍 봉 희

e-mail : bhhong@pusan.ac.kr

1982년 서울대학교 전자계산기공학과 졸업(학사)

1984년 서울대학교 대학원 전자계산기공학과 졸업(공학석사)

1988년 서울대학교 대학원 전자계산기공학과 졸업(공학박사)

1987년~현 재 부산대학교 컴퓨터공학과 교수

관심분야 : 공간 데이터베이스, RFID 시스템, RFID 데이터베이스

반 재 훈

e-mail : chban@kit.ac.kr

1997년 부산대학교 컴퓨터공학과 (학사)

1999년 부산대학교 대학원 컴퓨터공학과 (공학석사)

2006년 부산대학교 대학원 컴퓨터공학과 (공학박사)

2002년~현 재 경남정보대학 인터넷 응용계열 교수

관심분야 : 데이터베이스, 이동체 데이터베이스, RFID 시스템