USN 환경을 위한 공간 센서 데이타베이스 시스템의 설계 및 구현

Design and Implementation of a Spatial Sensor Database System for the USN Environment

신 인 수[*] 유 뢰[**] 김 정 준[***] 장 태 수[****] 한 기 준[*****]
In-Su Shin Lei Liu Joung-Joon Kim Tae-Soo Chang Ki-Joon Han

요 약 일반적으로 비공간 센서 데이타뿐만 아니라 공간 센서 데이타를 동시에 사용하는 유비쿼터스 센서 네트워크 (USN) 환경을 위해서는 이러한 센서 데이타를 효율적으로 관리하는 센서 데이타베이스 시스템은 매우 중요하다. 이러한 이유에서 Tiny DB, Cougar 등과 같은 다양한 센서 데이타베이스 시스템들이 연구되고 있다. 그러나, 이들 대부분은 공간 센서 데이타 처리를 위한 공간 데이타 타입과 공간 연산자를 지원하지 않기 때문에 공간 센서 데이타를 처리하는데 어려움이 존재한다. 그러므로, 본 논문은 USN 환경에서 공간 센서 데이타에 대한 효율적인 관리를 위해 TinyDB를 확장하여 공간 센서 데이타베이스 시스템을 설계 및 구현하였다. 특히 공간 센서 데이타베이스 시스템은 공간 데이타 타입과 공간 연산자를 지원하고, 공간 데이타 스트림으로 인해 발생하는 시스템 부하를 줄이기 위해서 메모리 관리 기능과 필터링 기능 등을 제공한다. 마지막으로, 수행 시간, 정확도, 메모리 사용량 등 다양한 성능 평가를 통해 공간 센서 데이타베이스 시스템이 기존 TinyDB에 비해 성능이 우수하다는 것을 증명하였다.

키워드 : USN, 센서 데이타베이스 시스템, 공간 데이타 타입, 공간 연산자, TinyDB

Abstract For the USN(Ubiquitous Sensor Network) environment which generally uses spatial sensor data as well as aspatial sensor data, a sensor database system to manage these sensor data is essential. In this reason, some sensor database systems such as TinyDB, Cougar are being developed by many researchers. However, since most of them do not support spatial data types and spatial operators to manage spatial sensor data, they have difficulty in processing spatial sensor data. Therefore, this paper developed a spatial sensor database system by extending TinyDB. Especially, the system supports spatial data types and spatial operators to TinyDB in order to manage spatial sensor data efficiently and provides the memory management function and the filtering function to reduce the system overload caused by sensor data streams. Lastly, we compared the processing time, accuracy, and memory usage of the spatial sensor database system with those of TinyDB and proved its superiority through the performance evaluation.

Keywords : USN, Sensor Database System, Spatial Data Types, Spatial Operators, TinyDB

1. 서 론

최근 온도, 습도, 압력 등과 같이 다양한 데이타를 센싱하는 센서 기술과 CDMA, WiFi, WiBro 등과 같은 무선 통신 기술이 발전함에 따라 생태계 모니터링, 홈 자동화, 차량 도난 감지 등과 같은 응용 분야에서 유비쿼터스 센서 네트워크 (USN : Ubiquitous Sensor Networks) 관련 기술에 대한 관심과 연구가 증대되고 있다[4].

USN이란 주로 무선 통신 방식으로 연결된 다양

† 이 논문은 2011학년도 건국대학교의 연구년교원 지원에 의해서 연구되었음.

* 건국대학교 컴퓨터공학부 박사과정 isshin@db.konkuk.ac.kr
** China Banking Regulatory Commission Financial Regulator dancingmokey68@gmail.com
*** 건국대학교 컴퓨터공학부 강의교수 jjkim9@db.konkuk.ac.kr
**** 비츠로시스 회장 vchairman@vitzro.com
***** 건국대학교 컴퓨터공학부 교수 kjhan@db.konkuk.ac.kr(교신저자)

한 종류의 센서 노드들이 센싱한 데이타를 관리하는 통신망이다[1, 15]. 센서 노드는 센싱, 프로세싱, 저장, 통신 모듈로 구성되며 각 센서 노드에는 하드웨어와 소프트웨어가 제한적이다. 즉, 센싱된 데이타를 처리하기 위한 프로세서 능력, 센싱된 데이타를 저장하기 위한 기억 장소 용량, 센싱된 데이타를 전송할 수 있는 전송 거리, 센서 노드의 전원의 크기 등이 제한된다. 특히, 센서 노드의 전원은 데이타 처리보다 데이타 전송으로 인한 에너지 소모가 매우 크다[17].

특히 GeoSensor는 RFID 리더, GPS, CCTV 등을 통해 직접 또는 간접적으로 자신의 위치를 획득할 수 있으며, 또한 해당 위치와 연관된 다양한 형태의 값들을 스트림의 형태로 생성한다[11]. 특히, USN 환경에서는 u-GIS, u-LBS, u-물류, u-교통, u-의료, u-방재 등과 관련된 다양한 서비스를 제공하기 위해서 GeoSensor의 활용이 증가하고 있으며, 이러한 GeoSensor는 비공간 데이타와 공간 데이타를 동시에 사용하는 유비쿼터스 시대를 이끌어 가고 있다. 따라서 USN 환경에서 이러한 GeoSensor의 공간 센서 데이타를 효율적으로 관리하기 위한 연구가 활발히 진행되고 있다[13, 18].

최근, USN 환경에서 센서 데이타에 대한 효율적인 관리를 위해 TinyDB[9], Cougar[16] 등과 같은 다양한 센서 데이타베이스 시스템들이 연구되고 있다. 그러나, 기존 센서 데이타베이스 시스템들은 공간 데이타 타입과 공간 연산자를 지원하지 않기 때문에 GeoSensor의 공간 센서 데이타를 효율적으로 관리하기에 어려움이 있다. 또한 최근에 TinyDB를 공간적으로 확장한 SE TinyDB[3]는 자체적으로 공간 연산자를 제공하나 OGC에서 제시한 국제 표준을 지원하지 못하고 있다.

따라서, 본 논문에서는 USN 환경에서 공간 센서 데이타에 대한 효율적인 관리를 위해 다양한 공간 데이타 타입 및 공간 연산자를 제공하는 공간 센서 데이타베이스 시스템을 설계 및 구현하였다. 특히 본 논문에서 개발한 공간 센서 데이타베이스 시스템은 공간 센서 데이타의 효율적인 관리를 위해 기존 센서 데이타베이스 시스템인 TinyDB를 확장하여 개발되며, 상호운용성을 위하여 공간 데이타 타입과 공간 연산자 확장시 OGC에서 제시한 Simple Feature Specification for SQL 표준[12]을 사용였다.

본 논문의 구성은 다음과 같다. 제1장 서론에 이어 제2장에서는 관련 연구로서 기존의 센서 데이타베이스 시스템인 TinyDB, SE TinyDB에 대해서 분석한다. 제3장에서는 본 논문에서 개발한 공간 센서 데이타베이스 시스템의 전체 시스템 구조와 각 관리자에 대하여 설명한다. 제4장에서는 성능 평가를 통해 본 논문에서 개발한 공간 센서 데이타베이스 시스템의 우수성을 입증한다. 마지막으로 제5장에서는 결론에 대하여 언급한다.

2. 관련연구

본 장에서는 본 논문에서 구현 시 사용한 TinyDB에 대해서 분석하고, 또한 이를 공간적으로 확장한 SE TinyDB에 대해서 설명한다.

2.1 TinyDB

TinyDB[8, 9]는 TinyOS에서 무선 센서 네트워크로부터 정보를 추출하기 위한 질의 처리 시스템이다. TinyOS[7]는 현재 가장 널리 사용되고 있는 대표적인 센서 운영 체제로서 미국 버클리 대학에서 nesC 언어로 개발한 오픈 소스 시스템이다. TinyOS는 컴포넌트 단위로 시스템 모듈화를 구성하고, 각 컴포넌트는 다른 컴포넌트들과 인터페이스라는 함수 호출을 통해 동작을 연결한다. 그러므로, 애플리케이션 개발자들은 컴포넌트들을 라이브러리로 사용하고, 또한 인터페이스들을 사용해서 컴포넌트들을 서로 연결하여 애플리케이션을 개발할 수 있다.

TinyDB의 특징은 다음과 같다. 첫째, 센서 네트워크에서 받은 센서의 종류를 기술하기 위해 메타데이타 카탈로그를 제공한다. 둘째, 사용자가 쉽게 원하는 데이타를 기술하는 질의 언어를 지원한다. 셋째, 네트워크 망을 가지고 있다. 넷째, 동시에 여러 센서 노드들에게 다중 질의가 가능하다. 다섯째, TinyDB 센서 네트워크를 확장하기 위해서 단순히 새로운 노드에 표준 TinyDB 코드를 다운로드 하고 리셋만으로 가능하다.

TinyDB에서 질의 처리 과정은 다음과 같다. 우선 서버 PC에서 입력받은 질의는 네트워크에 최적화된 질의로 전파된다. 그리고 전파된 질의를 수신한 노드는 라우팅 트리를 기반으로 이웃 노드로부터 데이타를 취득한 후 네트워크 안에서 집계 연산을 수행한다. 마지막으로, 최종 결과는 서버 PC로

전달된다.

사용자가 직접 프로그래밍을 하지 않고 원하는 데이타를 추출하기 위해서 TinyDB는 SQL과 유사한 간단한 인터페이스를 제공한다. 표 1은 TinyDB에서 제공된 SQL 유사 인터페이스를 지원하는 질의 구문 형식 및 질의 예를 보여준다.

표 1. 질의 구문 형식 및 질의 예

질의 구문 형식	SELECT select-list <FROM sensors> WHERE where-clause <GROUP BY gb-list> <HAVING having-list> <TRIGGER ACTION command-name<(param)>> <EPOCH DURATION>
질의 예	SELECT AVG(temp) FROM sensors WHERE temp>100

표 1에서 보는 바와 같이 질의 구문 형식에서 SELECT, WHERE, GROUP BY, HAVING 절은 표준 SQL의 기능과 유사하다. TRIGGER ACTION은 질의의 결과가 WHERE 절을 만족시킬 때마다 소리, LED 등 트리거 액션을 수행한다. EPOCH DURATION은 Epoch 사이의 시간을 지정하기 위해 사용되며, 지속 시간 단위는 밀리초(ms)이다. 표 1의 질의 예는 온도가 100°를 초과하는 센서 노드들의 평균 온도를 구하는 질의이다.

표 2는 TinyDB가 지원하는 데이타 타입과 연산자를 보여준다.

표 2. 데이타 타입과 연산자

데이타 타입	연산자
int 8, int 16, int 32, unit8, unit16, unit32, String	SUM, AVERAGE, MIN, MAX, COUNT

표 2에서 보는 바와 같이 TinyDB에서는 공간 데이타 타입과 공간 연산자를 지원하지 않기 때문에 공간 데이타를 효율적으로 관리하기 어려운 단점을 갖고 있다.

2.2 SE TinyDB

센서 노드에서 공간 질의와 비공간 질의를 모두 처리하기 위해서 TinyDB를 확장한 SE TinyDB가 설계 및 개발되었다[3]. SE TinyDB는 기존의 TinyDB에 공간 연산자 DISTANCE, INBOX, BEYONDBOUNDARY를 추가하여 확장한 시스템이다. 표 3은 SE TinyDB에서 추가된 공간 연산자를 보여준다.

표 3. SE TinyDB에서 추가된 공간 연산자

연산자	설명
DISTANCE	DISTANCE<ID, X, Y>
INBOX	INBOX<Xmin, Ymin, Xmax, Ymax>
BEYOND BOUNDARY	BEYONDBOUNDARY<Xmin, Ymin, Xmax, Ymax, CMD(par)>

표 3에서 보는 바와 같이 DISTANCE 연산자는 센서 노드들 간의 좌표 거리를 반환하고, INBOX 연산자는 특정 사각형 안에 포함되는 센서 노드의 위치 데이타를 반환한다. 그리고, BEYONDBOUNDARY 연산자는 특정 사각형의 경계를 벗어나는 센서 노드의 위치 데이타를 반환한다.

표 4는 SE TinyDB에서 공간 연산자를 사용한 공간 질의 예를 보여준다.

표 4. 공간 연산자를 사용한 공간 질의 예

공간 질의 예1	SELECT nodeid, temp, lat, lon FROM sensors WHERE DISTANCE[lat, lon, 500, 500] < 200
공간 질의 예2	SELECT nodeid, temp, lat, lon FROM sensors WHERE INBOX[0, 0, 500, 500]
공간 질의 예3	SELECT nodeid, temp, lat, lon FROM sensors WHERE BEYONDBOUNDARY[0, 0, 500, 500]

표 4에서 보는 바와 같이 공간 질의 예 1은 센서 노드 위치와 특정 지점(Point(500 500))간의 거리가 200이내인 센서 노드의 아이디(nodeid), 온도(temp), 위치 좌표(lat, lon)를 반환한다. 공간 질의 예 2는 특정 사각형 (0, 0, 500, 500) 안에 포함되는 센서 노드의 아이디(nodeid), 온도(temp), 위치 좌표(lat, lon)를 반환한다. 마지막으로 공간 질의 예 3는 특정 사각형 (0, 0, 500, 500)의 경계를 벗어나는 센서 노드의 아이디(nodeid), 온도(temp), 위치 좌표(lat, lon)를 반환한다.

SE TinyDB에서는 표 4와 같이 공간 연산자 3개

를 추가해서 특정 구역이나 이동 궤적에 대한 질의 처리가 기능하도록 하였다. 그러나 SE TinyDB는 OGC에서 제시한 Simple Feature Specification for SQL 표준을 따라서 다양한 공간 데이타 타입과 공간 연산자를 지원하지 않은 단점을 갖고 있다.

3. 공간 센서 데이타베이스 시스템

본 장에서는 공간 센서 데이타베이스 시스템의 전체 구조 및 공간 센서 데이타베이스 시스템을 구성하는 관리자들에 대해서 설명한다.

3.1 전체 시스템 구조

공간 센서 데이타베이스 시스템은 USN 환경에서 공간 센서 데이타를 효율적으로 관리할 수 있도록 TinyDB를 확장한 센서 데이타베이스 시스템이다. 그림 1은 본 논문에서 개발한 공간 센서 데이타베이스 시스템의 전체 구조를 보여준다.

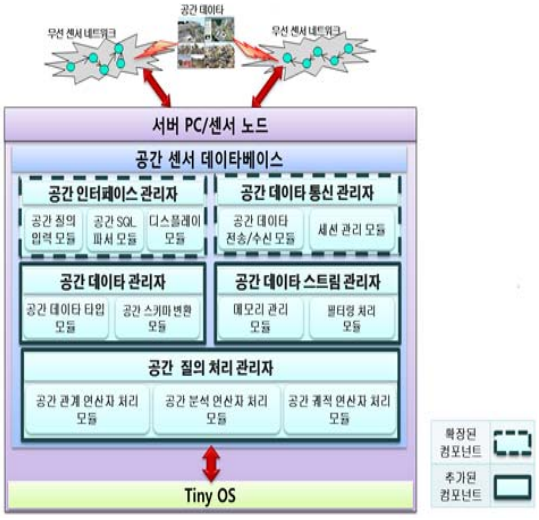

그림 1. 공간 센서 데이타베이스 시스템의 구조

그림 1에서 보는 바와 같이 공간 센서 데이타베이스 시스템은 공간 데이타 관리자, 공간 질의 처리 관리자, 공간 데이타 스트림 관리자, 공간 인터페이스 관리자, 공간 통신 관리자로 구성된다. 그림 1에서 점선 박스는 기존 TinyDB 컴포넌트를 확장한 컴포넌트이고, 실선 박스는 TinyDB에 추가된 컴포넌트이다.

공간 데이타 관리자는 OGC 표준을 따르는 공간 데이타 타입을 관리하고 공간 속성 정보를 공간 스키마로 변환한다. 그리고 공간 질의 처리 관리자는 OGC 표준을 따르는 공간 관계 연산자와 공간 분석 연산자를 제공하고, 또한 이동 센서 노드의 이동 궤적을 처리하기 위해 공간 궤적 연산자를 제공한다. 공간 데이타 스트림 관리자는 공간 질의를 처리할 때 공간 질의들 간의 메모리를 공유하는 기능과 입력 부하를 줄이기 위해 필터링하는 기능을 제공한다. 그리고 공간 인터페이스 관리자는 공간 질의를 입력받고 공간 SQL을 파싱한 후에 최종 공간 질의 결과를 디스플레이하는 기능을 제공한다. 마지막으로, 공간 데이타 통신 관리자는 서버 PC와 무선 센서 네트워크의 센서 노드 간, 그리고 각 센서 노드 간의 통신과 세션을 관리한다.

3.2 공간 데이타 관리자

본 절에서는 공간 데이타 관리자를 구성하는 공간 데이타 타입 관리 모듈, 공간 스키마 모듈에 대해서 설명한다.

3.2.1 공간 데이타 타입 관리 모듈

공간 데이타 타입 관리 모듈은 입력받은 공간 질의를 지원하기 위해 OGC의 “Simple Features Specification for SQL” 표준 명세[12]에서 제시하는 공간 데이타 타입을 제공한다. 표 5는 공간 데이타 타입 관리 모듈에서 지원하는 공간 데이타 타입과 사용 예를 보여준다.

표 5. 공간 데이타 타입과 사용 예

공간 데이타 타입	사용 예
Point	POINT (10 10)
LineString	LINESTRING (10 10, 20 20, 30 40)
Polygon	POLYGON (10 10, 10 20, 20 20, 20 15, 10 10)
PolyhedralSurface	POLYHEDRALSURFACE ((10 10, 10 20, 20 20, 20 10, 10 10), (20 10, 40 20, 20 20, 20 10))
MultiPoint	MULTIPOINT (10 10, 20 20)
MultiLineString	MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))
MultiPolygon	MULTIPOLYGON ((10 10, 10 20, 20 20, 20 15, 10 10),(60 60, 70 70, 80 60, 60 60))

표 5에서 보는 바와 같이 공간 데이타 타입 관리 모듈은 Point, LineString, Polygon, PolyhedralSurface, MultiPoint, MultiLineString, MultiPolygon 등 7개 공간 데이타 타입을 지원한다.

3.2.2 공간 스키마 변환 모듈

공간 스키마 변환 모듈은 공간 데이타 통신 관리자로부터 전달받은 공간 속성 정보를 공간 스키마 매핑 룰에 따라 공간 스키마로 변환하는 기능을 수행한다. 그림 2는 공간 스키마 변환 과정을 보여준다.

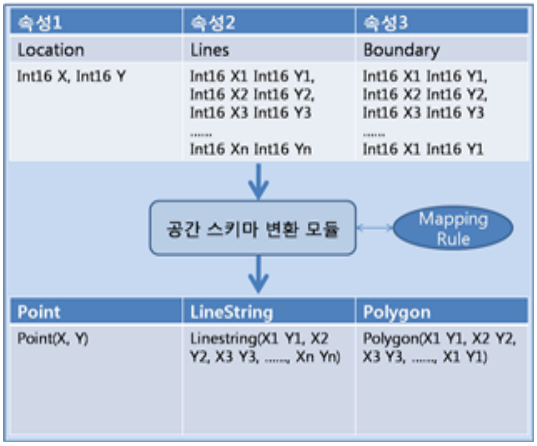

그림 2. 공간 스키마 변환 과정

그림 2에서 보는 바와 같이 공간 스키마 변환 모듈은 공간 속성 정보 Location, Lines, Boundary 등을 입력받아서 각각 공간 스키마 매핑 룰에 따라 Point, LineString, Polygon 등으로 변환한다.

3.3 공간 질의 처리 관리자

본 절에서는 공간 질의 처리 관리자를 구성하는 공간 관계 연산자 처리 모듈, 공간 분석 연산자 처리 모듈, 공간 궤적 연산자 처리 모듈에 대해서 설명한다.

3.3.1 공간 관계 연산자 처리 모듈

공간 관계 연산자 처리 모듈은 서버 PC와 센서 노드에서 공간 질의 처리를 위해 OGC의 “Simple Features Specification for SQL” 표준 명세[12]에서 제시하는 공간 관계 연산자를 제공한다. 표 6은 공간 관계 연산자 처리 모듈에서 제공하는 공간 관계 연산자를 보여준다.

표 6. 공간 관계 연산자

공간 관계 연산자	설 명
Equals(Geometry A, Geometry B)	객체 A와 객체 B가 동일한지 여부 반환
Disjoint(Geometry A, Geometry B)	객체 A와 객체 B가 떨어져 있는지 여부 반환
Touches(Geometry A, Geometry B)	객체 A와 객체 B의 경계가 만나는지 여부 반환
Within(Geometry A, Geometry B)	객체 A가 객체 B에 포함되는지 여부 반환
Overlaps(Geometry A, Geometry B)	객체 A와 객체 B가 겹치는지 여부 반환
Crosses(Geometry A, Geometry B)	객체 A와 객체 B가 교차하는지 여부 반환
Intersects(Geometry A, Geometry B)	객체 A와 객체 B가 만나는지 여부 반환
Contains(Geometry A, Geometry B)	객체 A가 객체 B를 포함하는지 여부 반환

표 6에서 보는 바와 같이 공간 관계 연산자 처리 모듈은 Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects, Contains 등 8개 공간 관계 연산자를 지원하는데, 공간 관계 연산자는 입력 값으로 두 개의 공간 객체 Geometry A, Geometry B를 입력받아 결과 값으로 True 또는 False를 반환한다.

3.3.2 공간 분석 연산자 처리 모듈

공간 분석 연산자 처리 모듈은 서버 PC와 센서 노드에서 공간 질의 처리를 위해 OGC의 “Simple Features Specification for SQL” 표준 명세[12]에서 제시하는 공간 분석 연산자를 제공한다. 표 7은 공간 분석 연산자 처리 모듈에서 제공하는 공간 분석 연산자를 보여준다.

표 7에서 보는 바와 같이 공간 분석 연산자 처리 모듈은 Distances, Intersection, Difference, Union, Buffer, ConvexHull 등 6개 공간 분석 연산자를 지원하는데, 공간 분석 연산자는 입력 값으로 두 개의 공간 객체 Geometry A, Geometry B를 입력받아 결과 값으로 새로운 공간 객체를 반환한다.

3.3.3 공간 궤적 연산자 처리 모듈

공간 궤적 연산자 처리 모듈은 센서 노드의 이동 궤적을 처리하기 위해 공간 궤적 연산자를 제공한다[2, 6, 14]. 표 8은 공간 궤적 연산자 처리 모듈에

표 7. 공간 분석 연산자

공간 분석 연산자	설 명
Distance(Geometry A, Geometry B)	객체 A와 객체 B의 거리를 반환
Intersection(Geometry A, Geometry B)	객체 A와 객체 B의 교집합을 반환
Difference(Geometry A, Geometry B)	객체 A와 객체 B의 차집합을 반환
Union(Geometry A, Geometry B)	객체 A와 객체 B의 합집합을 반환
Buffer(Geometry A, Double L)	객체 A의 경계로부터 L만큼 확장된 공간 객체를 반환
ConvexHull (Geometry A)	객체 A를 포함할 수 있는 가장 작은 볼록 다각형을 반환

서 제공하는 공간 궤적 연산자를 보여준다.

표 8에서 보는 바와 같이 공간 궤적 연산자 처리 모듈은 Enter, Insides, Leaves, Meets, Passes 등 5개 공간 궤적 연산자를 지원하는데, 공간 궤적 연산자는 입력 값으로 두 개의 공간 객체 Geometry A, Geometry B를 입력받아 결과 값으로 True 또는 False를 반환한다.

표 8. 공간 궤적 연산자

공간 궤적 연산자	설 명
Enter(Geometry A, Geometry B)	객체 A가 객체 B의 외부에서 내부로 들어가는지 여부 반환
Insides(Geometry A, Geometry B)	객체 A가 객체 B의 내부에서만 존재하는지 여부 반환
Leaves(Geometry A, Geometry B)	객체 A가 객체 B의 내부에서 외부로 나가는지 여부 반환
Meets(Geometry A, Geometry B)	객체 A가 객체 B의 경계만 터치하는지 여부 반환
Passes(Geometry A, Double L)	객체 A가 객체 B의 외부에서 내부로 들어왔다가 다시 외부로 나가는지 여부 반환

3.4 공간 데이타 스트림 관리자

본 절에서는 공간 데이타 스트림 관리자를 구성하는 메모리 관리 모듈, 필터링 처리 모듈에 대해서 설명한다.

3.4.1 메모리 관리 모듈

메모리 관리 모듈은 공간 센서 데이타베이스 시스템에서 실행되는 다양한 공간 질의들 간에 필요한 데이타를 공유하도록 하여 공간 데이타 스트림으로 인해 발생하는 시스템 부하를 줄이는 기능을 제공한다[5, 13].

공간 센서 데이타베이스 시스템에서 다양한 공간 질의가 동시에 수행될 때 공간 질의들은 각자의 메모리 영역을 갖지 않고 하나의 메모리 영역을 공유하게 된다. 이 때 공유된 메모리에는 각 데이타 튜플에 대한 참조 카운터가 설정된다. 공간 질의가 한 데이타 튜플을 처리하게 되면 해당 데이타 튜플의 참조 카운터가 1이 감소되고, 참조 카운터가 0이 되면 해당 데이타 튜플이 메모리에서 삭제된다.

예를 들어, 공간 질의 1이 특정 튜플을 저장소에 저장하고, 공간 질의 2와 3은 이를 공유한 경우에는 그 튜플의 참조 카운터가 3이 된다. 공간 질의 1이 그 튜플을 처리하면 저장소에서 그 튜플은 바로 삭제되지 않고, 참조 카운터만 1을 줄인다. 공간 질의 2와 3이 그 튜플을 모두 처리하면 참조 카운터가 0이 되면서 그 튜플은 삭제된다.

3.4.2 필터링 처리 모듈

필터링 처리 모듈은 정확도 손실을 최소화 하면서 입력 데이타 스트림 양을 줄여 과부하 문제를 해결하기 위해 필터링 기능을 제공한다[10, 17]. 즉, 필터링 처리 모듈은 위치 좌표 거리 차이, 특정 시간에 따른 시간 범위, 특정 센서 노드들의 아이디 등으로 구성된 필터링 조건을 사용하여 입력 데이타 스트림에 대해 필터링을 수행하고, 필터링된 데이타 스트림만을 질의 처리 관리자에게 전달한다. 그림 3은 센서 노드들의 아이디와 위치 좌표 거리 차이로 구성된 필터링 조건에 따라 공간 데이타가 필터링되는 예를 보여준다.

그림 3에서 보는 바와 같이 설정된 필터링 조건이 아이디가 1부터 5까지이고 위치 좌표 거리 차이

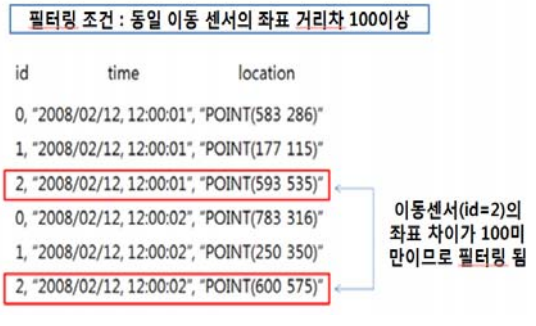

그림 3. 좌표 거리 차이를 사용한 필터링 예

가 100 이상이라면 센서 노드는 수집되는 데이타 스트림으로부터 아이디가 1부터 5까지이고 동일 객체의 이전 위치와 현재 위치의 거리 차이가 100 이상인 데이타만을 질의 처리 관리자에게 전해준다.

3.5 공간 인터페이스 관리자

본 절에서는 공간 인터페이스 관리자를 구성하는 공간 질의 입력 모듈, 공간 SQL 파서 모듈, 디스플레이 모듈에 대해서 설명한다.

3.5.1 공간 질의 입력 모듈

공간 질의 입력 모듈은 사용자로부터 공간 질의를 입력받아 공간 SQL 파서 모듈로 전달하는 기능을 수행한다. 또한 공간 질의 결과에 대해 텍스트/그래픽 디스플레이 모드를 선택하는 기능과 공간 질의의 각 파라미터 값(수행 주기, 질의 ID, 질의 조건 등)을 입력하는 기능을 제공한다.

공간 데이타베이스 시스템에서는 공간 질의 구문 형식은 TinyDB 경우와 동일하나 공간 질의를 작성할 때 공간 데이타 타입뿐만 아니라 공간 관계 연산자, 공간 분석 연산자, 공간 궤적 연산자를 사용하는 것이 가능하다. 표 9는 TinyDB에서는 불가능하지만 공간 센서 데이타베이스 시스템에서는 가능한 공간 질의 예를 보여준다.

표 9. 공간 질의 예

질의 종류	공간 질의 예
공간 관계 연산자를 사용한 질의 예	SELECT nodeid, temp, loc FROM sensors WHERE Contains(loc, polygon(0 0, 40 0, 40 40, 0 40))
공간 분석 연산자를 사용한 질의 예	SELECT nodeid, temp, loc FROM sensors WHERE Contains(loc, Intersection(Polygon(0 0, 0 40, 40 40, 40 0, 0 0), Polygon(30 30, 30 70, 70 70, 70 30, 30 30))
공간 궤적 연산자를 사용한 질의 예	SELECT nodeid, temp, loc FROM sensors WHERE Passes(loc, Polygon(0 0, 0 40, 40 40, 40 0, 0 0))

표 9에서 보는 바와 같이 공간 센서 데이타베이스 시스템에서는 공간 데이타 타입, 공간 관계 연산자, 공간 분석 연산자, 공간 궤적 연산자를 사용하여 다양한 공간 질의를 작성할 수 있다.

3.5.2 공간 SQL 파서 모듈

공간 SQL 파서 모듈은 공간 질의 입력 모듈로부터 입력받은 공간 SQL 구문을 파싱하고(즉, 어휘 분석과 구문 분석 수행) 파싱된 정보를 바탕으로 공간 SQL 구문의 오류 및 유효성을 검사하는 기능과 잘못된 공간 SQL 구문에 대해 오류 처리 기능을 제공한다. 또한 정상적으로 공간 SQL 파싱 작업이 완료된 공간 질의에 대한 공간 속성 및 공간 연산자 정보를 생성하여 공간 데이타 통신 관리자에 전달한다.

3.5.3 디스플레이 모듈

디스플레이 모듈은 서버 PC의 화면에 질의 실행 상황, 최종 질의 결과 등을 디스플레이하는 기능을 제공한다. 디스플레이 모듈을 통해 공간 질의 처리 관리자로부터 전달받은 최종 질의 결과를 텍스트와 그래프 2가지 형식으로 보여줄 수 있으며, 공간 SQL 파서 모듈로부터 전달받은 공간 SQL 질의문을 디스플레이할 수 있다. 또한 공간 질의 수행 과정 중 발생된 오류에 대한 오류 메시지를 디스플레이하는 기능도 제공한다. 마지막으로 공간 질의 수행 중에 질의 수행 주기를 재설정할 수 있으며, 공간 질의를 일시 중지, 재시작 등의 동작도 제어할 수 있다.

3.6 공간 데이타 통신 관리자

본 절에서는 공간 데이타 통신 관리자를 구성하는 세션 관리 모듈, 공간 데이타 전송/수신 모듈에 대해서 설명한다.

3.6.1 세션 관리 모듈

세션 관리 모듈은 서버 PC와 센서 노드 간, 그리고 무선 센서 네트워크내의 각 센서 노드 간의 세션을 생성, 유지, 삭제하는 기능을 제공한다. 새로운 공간 질의를 시작할 때 세션을 생성하고, 세션의 유연한 관리를 위해 기존의 공간 질의가 끝날 때 해당 세션을 삭제한다. 또한 질의 수행 주기를 설정한 경우에는 세션의 유효 시간을 재설정하는 기능도 제공한다.

3.6.2 공간 데이타 전송/수신 모듈

공간 데이타 전송/수신 모듈은 서버 PC와 센서 노드 간, 그리고 무선 센서 네트워크내의 각 센서 노드 간의 데이타 전송/수신 기능을 제공한다. 서버

PC측의 공간 데이타 전송/수신 모듈은 공간 인터페이스 관리자로부터 전달받은 공간 속성 및 공간 연산자 정보를 무선 센서 네트워크의 각 센서 노드에게 전송하고, 무선 센서 네트워크로부터 전달받은 공간 질의 최종 결과를 공간 인터페이스 관리자로 전송한다. 센서 노드측의 공간 데이타 전송/수신 모듈은 다른 센서 노드로부터 전달받은 공간 데이타를 질의 처리 관리자로 전송하고, 해당 센서 노드가 센싱한 공간 데이타를 다른 센서 노드에게 전송한다.

특히 전송할 공간 데이타가 제한 크기를 초과하는 경우에는 공간 데이타를 분할하여 전송하며, 또한 공간 데이타 전송이 실패하는 경우에는 자동으로 재전송한다. 그리고 센서 노드의 제한된 자원에 맞추어 효율적인 공간 데이타 전송/수신 동작 방식을 제공하는데, 그림 4는 공간 데이타 전송/수신 과정을 보여준다.

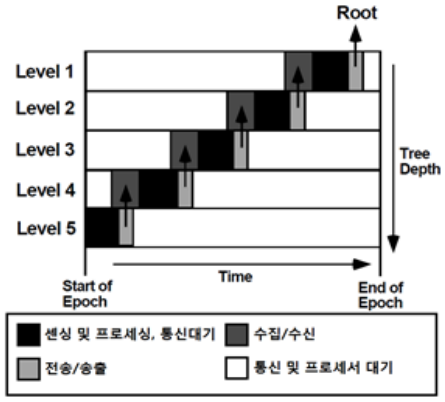

그림 4. 공간 데이타 전송/수신 과정

그림 4에서 보는 바와 같이 센서 노드들은 전원을 절약하기 위해 대부분 대기 상태에 존재하고 있으며, 또한 데이타의 센싱, 수신, 전송 등의 동작이 정해진 시간 만큼 수행된 후에는 다시 자동으로 대기 상태로 전환된다.

4. 성능 평가

본 장에서는 본 논문에서 개발한 공간 센서 데이타베이스 시스템과 TinyDB의 수행 시간, 정확도, 메모리 사용량에 대한 성능 평가 결과에 대해 분석한다. 특히 모든 성능 평가에서 공간 센서 데이타베이스 시스템에 대해서는 공간 데이타 필터링을 사용했을 경우와 사용하지 않았을 경우로 구분하여 성능 평가를 수행하였다.

4.1 성능 평가 환경

본 논문에서는 공간 센서 데이타베이스 시스템을 구현하기 위하여 운영체제로는 Cygwin 2.5.7하의 TinyOS 1.1.15 환경을 사용하였으며, 개발 도구로는 TinyOS에서 제공하는 nesC 1.2.8 버전과 g++ 3.4.3 버전을 사용하였다. 또한 자바 GUI는 Windows XP Professional 환경을 기반으로 하였으며, 개발 도구로는 JAVA 1.4 버전을 사용하였다.

특히 성능 평가를 수행하기 위해 공간 질의 수행 주기, 1회 데이타 전송 크기, 센서 노드 개수, 동시 수행되는 공간 질의 개수 등의 파라미터 값을 설정하였다. 표 10은 성능 평가를 위해 설정된 파라미터 값과 사용된 질의 예를 보여준다.

표 10. 성능 평가를 위한 파라미터 값과 질의 예

파라미터	값	
공간 질의 수행 주기	256ms	
데이타 전송 크기	25Byte	
센서 노드 개수	1000개, 2000개, 3000개, 4000개	
공간 질의 개수	20개, 40개, 80개, 120개	
필터링 조건	좌표 거리 차이가 100이상	
질의 예	TinyDB	공간 센서 데이타베이스 시스템
	SELECT nodeid, temp, x, y FROM sensors WHERE temp>40 AND x>0 AND x<400 AND y>0 AND y<400	SELECT nodeid, temp, Loc FROM sensors WHERE Contains (Polygon(0 0, 0 400, 400 400, 400 0, 0 0), Loc)

4.2 수행 시간 성능 평가

그림 5는 센서 노드 개수에 따른 공간 센서 데이타베이스 시스템과 TinyDB의 수행 시간을 측정한 결과를 보여준다.

그림 5(a)에서 보는 바와 같이 필터링을 사용하지 않았을 경우에는 공간 센서 데이타베이스 시스템은 TinyDB에 비해 평균적으로 12% 정도 수행 시간이

빠른 성능을 보여준다. 이러한 성능 평가 결과는 공간 센서 데이타베이스 시스템에서는 공간 연산자가 지원되어 공간 질의가 빨리 처리될 수 있었기 때문이다. 그리고 그림 5(b)에서 보는 바와 같이 필터링을 사용했을 경우에는 공간 센서 데이타베이스 시스템은 TinyDB에 비해 평균적으로 21% 정도의 수행 시간이 빠른 성능을 보여준다. 그 이유는 공간 센서 데이타베이스 시스템에서는 입력 데이타에 대해 필터링을 수행하여 시스템 부하가 감소되었기 때문이다.

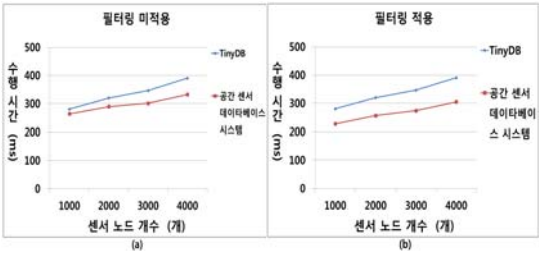

그림 5. 센서 노드 개수에 따른 수행 시간 성능 평가

4.3 정확도 성능 평가

그림 6은 센서 노드 개수에 따른 공간 센서 데이타베이스 시스템과 TinyDB의 정확도를 측정한 결과를 보여준다.

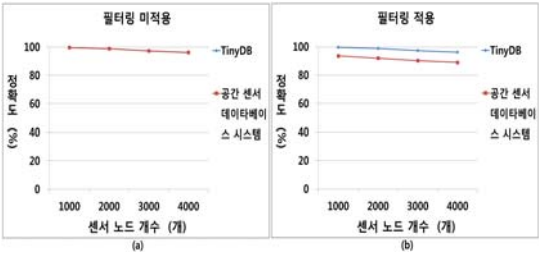

그림 6. 센서 노드 개수에 따른 정확도 성능 평가

그림 6(a)에서 보는 바와 같이 필터링을 사용하지 않았을 경우에는 공간 센서 데이타베이스 시스템은 TinyDB의 정확도와 동일한 성능을 보여준다. 그 이유는 두 경우 모두에서 입력 데이타의 손실이 없었기 때문이다. 그리고 그림 6(b)에서 보는 바와 같이 필터링을 사용했을 경우에는 공간 센서 데이타베이스 시스템은 TinyDB에 비해 평균적으로 7% 정도 정확도가 낮은 성능을 보여준다. 이러한 성능 평가 결과는 공간 센서 데이타베이스 시스템에서는 입력 데이타의 필터링으로 인해 일부 데이타의 손실을 야기하였기 때문이다.

4.4 메모리 사용량 성능 평가

그림 7은 센서 노드 개수에 따른 공간 센서 데이타베이스 시스템과 TinyDB의 메모리 사용량을 측정한 결과를 보여준다.

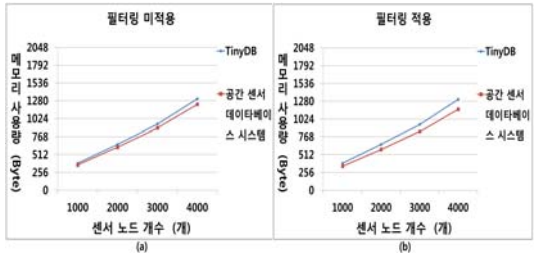

그림 7. 센서 노드 개수에 따른 메모리 사용량 성능 평가

그림 7(a)에서 보는 바와 같이 필터링을 사용하지 않았을 경우에는 공간 센서 데이타베이스 시스템은 TinyDB에 비해 평균적으로 6% 정도 메모리 사용량이 적은 성능을 보여준다. 이러한 성능 평가 결과는 공간 센서 데이타베이스 시스템에서는 여러 개 공간 질의 사이에서 메모리가 공유되었기 때문이다. 그리고 그림 7(b)에서 보는 바와 같이 필터링을 사용했을 경우에는 공간 센서 데이타베이스 시스템은 TinyDB에 비해 평균적으로 11% 정도 메모리 사용량이 적은 성능을 보여준다. 그 이유는 공간 데이타 스트림 관리자에서는 메모리 공유뿐만 아니라 입력 데이타의 필터링으로 인해 입력 데이타 스트림의 양을 줄였기 때문이다.

5. 결 론

본 논문은 USN 환경에서 공간 센서 데이타에 대한 효율적인 관리를 위해 기존의 TinyDB를 확장하여 공간 센서 데이타베이스 시스템을 설계 및 개발하였다. 공간 센서 데이타베이스 시스템은 국제 표준을 준수하는 공간 데이타 타입, 공간 관계 연산자, 공간 분석 연산자, 공간 궤적 연산자를 지원한다. 특히, 공간 센서 데이타베이스 시스템은 다양한 공간 질의들 간에 필요한 데이타를 공유하기 위해

서 메모리 관리 기능을 제공하며, 정확도 손실을 최소화 하면서 입력 데이타 스트림 양을 줄여 과부하 문제를 해결하기 위해 필터링 기능을 제공한다.

그리고 성능 평가를 통해 필터링을 사용했을 경우에만 공간 센서 데이타베이스 시스템이 TinyDB에 비해 평균적으로 정확도가 약간 낮은 성능을 보여주었지만 나머지 모든 경우에서는 공간 센서 데이타베이스 시스템이 수행 시간, 메모리 사용량 측면에서 TinyDB 보다 우수한 성능을 보여 주었다.

참 고 문 헌

[1] Andreou, P., Zeinalipour-Yazti, D., Pamboris, A., and Samaras, G., 2011, "Optimized Query Routing Trees for Wireless Sensor Networks," Information Systems, vol.36, no.2, pp.267-291.

[2] Erwig, M., and Schneider, M., 1999, "Developments in Spatio-Temporal Query Languages," Proc. of 10th Int. Workshop on Database and Expert Systems Applications, pp.441-449.

[3] Felice, P.D., Lanni, M., and Pomante, L., 2010, "Design and Evaluation of a Spatial Extension of TinyDB for Wireless Sensor Networks," International Journal of Computers and Their Applications Manuscript, vol.17, no.3, pp.172-193.

[4] Inoue, M., 2009, "A Model and System Architecture for Ubiquitous Sensor Network Businesses," Innovations for Digital Inclusions, pp.1-8.

[5] Krizanovic, K., Galic, Z., and Baranovic, M., 2010, "Spatio-Temporal Data Streams: an Approach to Managing Moving Objects," Proc. of the 33rd Int. Convention on MIPRO, pp.744-749.

[6] Lee, J.H., An, K.H., and Park, J.H., 2005, "Design of Query Language for Location-Based Services," Web and Wireless Geographical Information Systems, pp.11 - 18.

[7] Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., and Culler, D., 2005, "TinyOS: An Operating System for Wireless Sensor Networks," Ambient Intelligence, pp.115-148.

[8] Levis, P., and Wei, H., 2006, TinyDB: Design, Code and Implementation, http://csl.stanford.edu/~pal/pubs/tiyos-programming.pdf.

[9] Madden, S.R., Franklin, M.J., Hellerstein, J.M., and Hong, W., 2005, "Tiny DB: an Acquisitional Query Processing System for Sensor Networks," ACM Transactions on Database Systems, Vol.30, No.1, pp.122-173.

[10] Maier, D., Tucker, P.A., and Garofalakis, M., 2005, "Filtering, Punctuation, Windows and Synopses," Stream Data Management, Chapter 3, pp.35-58.

[11] Nittel, S., Labrinidis, A., and Stefanidis, A., 2008, "Introduction to Advances in Geosensor Networks," GeoSensor Networks, pp.1-6.

[12] Open Geospatial Consortium, Inc, 2010, OpenGIS Implementation Specification for Geographic Information-Simple Feature Access- Part 1: Common Architecture, Version 1.2.1.

[13] Park, J., Kim, K., Ahn, S., and Hong, B., 2010, "Continuous Query Processing on Combined Data Stream: Sensor, Location and Identification," Proc. of 7th Int. Conf. on Information Technology, pp.518-522.

[14] Pfoser, D., Jensen, C.S., and Theodoridis, Y., 2000, "Novel Approaches in Query Processing for Moving Objects," Proc. of 26th Int. Conf. on Very Large Data Bases, pp.395-406.

[15] Taslidere, E., Cohen, F.S., and Reisman, F.K., 2011, "Wireless Sensor Networks-A Hands-On Modular Experiments Platform for Enhanced Pedagogical Learning," IEEE Transactions on Education, Vol.54, No.1, pp.24-33.

[16] Yao, Y., and Gehrke, J., 2002, "The Cougar Approach to In-Network Query Processing in Sensor Networks," SIGMOD Record, Vol. 31, No. 3, pp.9-18.

[17] 김정준, 신인수, 이기영, 한기준, 2011, "무선 센서 네트워크에서 효율적인 집계 질의 처리," 한국공간정보학회지, 제19권, 제3호, pp.95-106.

[18] 정원일, 신숭선, 백성하, 이연, 이동욱, 김경배, 이충호, 김주완, 배해영, 2009, “u-GIS 컴퓨팅을 위한 GeoSensor 데이타 스트림 처리 시스템,” 한국공간정보학회지, 제11권, 제1호, pp.9-16.

논문접수 : 2011.11.15
수 정 일 : 1차 2011.12.22 / 2차 2012.01.12
심사완료 : 2012.01.16

신 인 수

2006년 건국대학교 컴퓨터공학 공학사
2008년 건국대학교 대학원 공학석사
2008년~현재 건국대학교 컴퓨터공학 박사과정
관심분야는 시공간 데이타베이스, 모바일 데이타베이스, Geo Semantic Web

유 뢰

2006년 남경항공항천대학교(중국) 컴퓨터공학 공학사
2011년 건국대학교 대학원 공학석사
2011년~현재 China Banking Regulatory Commission Financial Regulator
관심분야는 USN, 공간 센서 데이타베이스, GIS, LBS

김 정 준

2003년 건국대학교 컴퓨터공학 공학사
2005년 건국대학교 대학원 공학석사
2010년 건국대학교 대학원 공학박사
2010년~현재 건국대학교 컴퓨터공학부 강의교수
관심분야는 공간 데이타베이스, 시공간 데이타베이스, GIS, LBS, 텔레매틱스, USN, Semantic Web

장 태 수

1979년 한양대학교 전기공학 공학사
1986년 일본 고오베대 계측공학 공학석사
1990년 일본 고오베대 시스템공학 공학박사
1985년~1990년 광명전기 기술연구소 소장
1998년~2005년 (주)비츠로시스 대표이사
2005년~현재 (주)비츠로시스 회장
2010년~2011년 대한전기학회 부회장
관심분야는 센서 데이타베이스, GIS, U-CITY

한 기 준

1979년 서울대학교 수학교육학 이학사
1981년 한국과학기술원(KAIST) 전산학과 공학석사
1985년 한국과학기술원(KAIST) 전산학과 공학박사
1985년~현재 건국대학교 컴퓨터공학부 교수
1990년 Stanford 대학 전산학과 Visiting Scholar
2000년~2002년 한국정보과학회 데이타베이스 연구회 운영위원장
2004년~2006년 한국공간정보시스템학회 회장
2004년~2008년 한국정보시스템감리사협회 회장
관심분야는 공간 데이타베이스, GIS, LBS, 텔레매틱스