



- (51) 국제특허분류:
H04L 12/24 (2006.01)
- (21) 국제출원번호: PCT/KR2016/000852
- (22) 국제출원일: 2016 년 1 월 27 일 (27.01.2016)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보:
10-2015-0059752 2015 년 4 월 28 일 (28.04.2015) KR
- (71) 출원인: 한국과학기술원 (KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY) [KR/KR]; 34141 대전시 유성구 대학로 291, Daejeon (KR).
- (72) 발명자: 현순주 (HYUN, Soon Joo); 34141 대전시 유성구 대학로 291, Daejeon (KR). 임종석 (LIM, Chong Sok); 34141 대전시 유성구 대학로 291, Daejeon (KR). 박민지 (PARK, Min Jee); 34141 대전시 유성구 대학로 291, Daejeon (KR).
- (74) 대리인: 양성보 (YANG, Sungbo); 06099 서울시 강남구 선릉로 125 길 14 삼성빌딩 2층 피앤티특허법률사무소, Seoul (KR).

- (81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

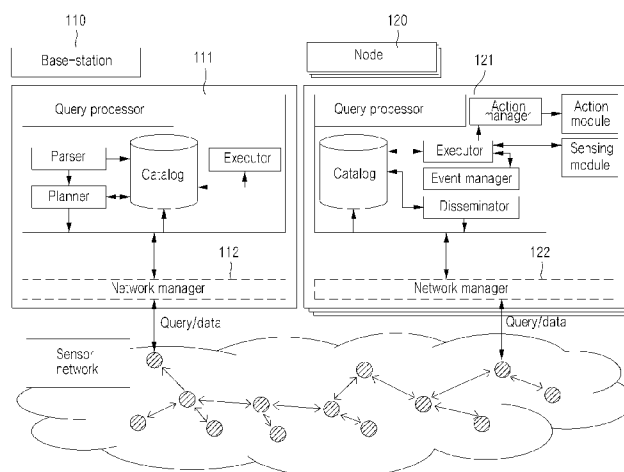
공개:

- 국제조사보고서와 함께 (조약 제 21 조(3))

(54) Title: OPERATOR FOR SPATIAL QUERY AND ENERGY EFFICIENT QUERY PROCESSING IN WIRELESS SENSOR NETWORK DATABASE SYSTEM

(54) 발명의 명칭 : 무선 센서 네트워크 데이터베이스 시스템에서 공간쿼리를 위한 연산자 및 에너지 효율적인 쿼리 처리

[도 11]



(57) Abstract: Disclosed is an operator for a spatial query and an energy efficient query processing technique in a wireless sensor network database system. A query processing method in a wireless sensor network comprises the steps of: transmitting, by a base station belonging to the wireless sensor network, a query on the wireless sensor network; and collecting, by a sensor node having received the query, sensor data according to the performance of the query, wherein the query is a requirement for data collection expressed in a sensor network query language (SNQL), and a spatial operator for identifying a geometric target space for performing the query is defined.

(57) 요약서:

[다음 쪽 계속]



무선 센서 네트워크 데이터베이스 시스템에서 공간쿼리를 위한 연산자 및 에너지 효율적인 쿼리 처리 기술이 개시된다. 무선 센서 네트워크에서의 쿼리 처리 방법은, 상기 무선 센서 네트워크에 속하는 기지국에서 상기 무선 센서 네트워크 상에 쿼리를 전송하는 단계; 및 상기 쿼리를 수신한 센서 노드에서, 상기 쿼리를 수행함에 따라 센서 데이터를 수집하는 단계를 포함하고, 상기 쿼리는 센서 네트워크 쿼리 언어(sensor network query language, SNQL)로 표현된 데이터 수집의 요구 사항으로, 상기 쿼리를 수행할 기하학적 타겟 영역(geometric target space)을 식별하기 위한 공간 연산자(spatial operator)가 정의된다.

명세서

발명의 명칭: 무선 센서 네트워크 데이터베이스 시스템에서 공간쿼리를 위한 연산자 및 에너지 효율적인 쿼리 처리

기술분야

- [1] 본 발명의 실시예들은 무선 센서 네트워크 데이터베이스 시스템에서 공간 쿼리를 처리하는 기술에 관한 것이다.

배경기술

- [2] 무선 센서 네트워크(Wireless Sensor Network: WSN)는 물리적 세계를 모니터링 하기 위해 다수의 센서 노드들로 구성된다. 무선 센서 네트워크(WSN)는 분산되어 설치된 센서 노드들로부터 데이터를 수집하여 사용한다(예컨대, Military applications, Environmental applications, Health care applications, Home applications, Other applications 등).
- [3] 무선 센서 네트워크(WSN)를 구성하는 센서 노드들로부터 데이터를 수집하는 가상의 저장 수단으로 센서 네트워크 데이터베이스(SNDB)를 활용한다.. 이러한 센서 네트워크 데이터베이스(SNDB)의 특징은 분산된 센서 노드들로부터 데이터를 수집한다는 점과 에너지 제한이 있는 디바이스로 구성된다는 점에 있다.
- [4] 기존 센서 네트워크 데이터베이스(SNDB)에서는 쿼리 언어의 불충분한 표현력(expressive power)으로 인해 노드들이 위치한 물리적 영역을 쿼리 조건으로 표현하기 어려운 문제가 있다. 또한, 멀티 타겟(multiple target)의 위치 파악을 위한 쿼리/데이터를 서버까지 전송해야 하므로 쿼리 처리를 위한 데이터 전송 비용이 커지고 노드의 에너지 소비가 커지는 문제가 발생한다. 그리고, 기존의 쿼리는 직사각형(rectangle) 영역을 대상으로 쿼리를 전송하기 때문에 비정형 영역 대상의 쿼리가 필요한 경우 불필요한 쿼리/데이터 전송이 발생하게 된다.
- [5] 이에, 센서 네트워크 데이터베이스(SNDB)에서 특정 영역을 대상으로 하는 쿼리 처리가 필요하며, 특정 영역을 대상으로 쿼리 처리 시 멀티 타겟에 대한 고려가 필요하다. 그리고, 센서 네트워크 데이터베이스(SNDB)에서 특정 영역 대상의 쿼리 처리 시 비정형 영역의 타겟에 대한 고려가 필요하다.

발명의 상세한 설명

기술적 과제

- [6] 쿼리를 위한 영역 정보를 용이하게 표현하고 처리할 수 있는 쿼리 처리 언어 및 효율적으로 쿼리를 처리할 수 있는 방법 및 장치를 제공한다.
- [7] 쿼리의 중간 결과 수집 후 이를 이용하여 쿼리를 다시 수행하는 경우 쿼리의 중간 결과를 서버까지 전송하지 않고 노드 레벨에서 처리하여 쿼리 처리 비용을 줄일 수 있는 방법 및 장치를 제공한다.

- [8] 센서 네트워크 데이터베이스(SNDB)에서 특정 영역 대상의 쿼리 처리 시 비정형 영역을 대상으로 쿼리를 처리할 수 있는 방법 및 장치를 제공한다.
- 과제 해결 수단**
- [9] 컴퓨터로 구현되는 무선 센서 네트워크에서의 쿼리 처리 방법에 있어서, 상기 무선 센서 네트워크에 속하는 기지국에서 상기 무선 센서 네트워크 상에 쿼리를 전송하는 단계; 및 상기 쿼리를 수신한 센서 노드에서, 상기 쿼리를 수행함에 따라 센서 데이터를 수집하는 단계를 포함하고, 상기 쿼리는 센서 네트워크 쿼리 언어(sensor network query language, SNQL)로 표현된 데이터 수집의 요구 사항으로, 상기 쿼리를 수행할 기하학적 타겟 영역(geometric target space)을 식별하기 위한 공간 연산자(spatial operator)가 정의되는 것을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법을 제공한다.
- [10] 일 측면에 따르면, 상기 공간 연산자는 주어진 조건(condition)을 만족시키는 기하학적 공간을 타겟 영역으로 설정하기 위한 엔벨롭 연산자(envelope operator)를 포함할 수 있다.
- [11] 다른 측면에 따르면, 상기 공간 연산자는 타겟 위치와 가장 가까운 기하학적 공간을 타겟 영역으로 설정하기 위한 니어바이 연산자(nearby operator), 주어진 기하학적 공간으로부터 주어진 거리 이내에 있는 공간을 타겟 영역으로 설정하기 위한 거리 연산자(distance operator), 주어진 기하학적 공간으로부터 주어진 방향에 있는 공간을 타겟 영역으로 설정하기 위한 방향 연산자(direction operator) 중 적어도 하나의 연산자를 더 포함할 수 있다.
- [12] 또 다른 측면에 따르면, 상기 공간 연산자는 둘 이상의 피연산자(operand)를 사용하는 집합 연산자(set operator)로서, 주어진 기하학적 공간 간의 교차 지역(intersection area)을 타겟 영역으로 설정하기 위한 교차 연산자(intersection operator), 주어진 기하학적 공간 간의 합집합 지역(union area)을 타겟 영역으로 설정하기 위한 합집합 연산자(union operator), 주어진 기하학적 공간 간의 차집합 지역(difference area)을 타겟 영역으로 설정하기 위한 차집합 연산자(difference operator) 중 적어도 하나의 연산자를 포함할 수 있다.
- [13] 또 다른 측면에 따르면, 상기 센서 데이터를 수집하는 단계는, 상기 쿼리가 둘 이상의 태스크를 포함하는 경우, 상기 쿼리를 전달하는 자식 노드의 수를 확인하는 단계; 상기 자식 노드의 수에 따라 LCA(lowest common ancestor) 노드 여부를 결정하는 단계; 상기 LCA 노드에 해당되는 경우 상기 자식 노드로부터 수신된 센서 데이터를 이용하여 상기 쿼리를 재구성하는 단계; 및 상기 재구성된 쿼리를 상기 자식 노드로 재전송하는 단계를 포함할 수 있다.
- [14] 또 다른 측면에 따르면, 상기 LCA 노드 여부를 결정하는 단계는, 상기 쿼리의 전송 경로에서 최초로 쿼리가 전달된 상기 자식 노드의 수가 1을 초과하는 경우 상기 LCA 노드로 결정할 수 있다.
- [15] 또 다른 측면에 따르면, 상기 센서 노드는 직접 연결된 적어도 하나의 자식

노드와 서브트리(subtree)의 메타데이터를 관리하고, 상기 자식 노드의 메타데이터는 상기 자식 노드의 좌표와 식별 번호 및 센서 종류를 포함하고, 상기 서브트리의 메타 데이터는 상기 서브트리의 MBR(minimum boundary rectangle)의 좌표와 센서 종류를 포함할 수 있다.

[16] 또 다른 측면에 따르면, 상기 서브트리의 MBR이 상기 타겟 영역에 해당되는 경우 상기 서브트리에 포함되는 자식 노드로 상기 쿼리가 전달될 수 있다.

[17] 적어도 하나의 프로그램이 로딩된 메모리; 및 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는, 상기 프로그램의 제어에 따라, 무선 센서 네트워크에 속하는 기지국에서 상기 무선 센서 네트워크 상에 쿼리를 전송하는 과정; 및 상기 쿼리를 수신한 센서 노드에서, 상기 쿼리를 수행함에 따라 센서 데이터를 수집하는 과정을 처리하고, 상기 쿼리는 센서 네트워크 쿼리 언어(sensor network query language, SNQL)로 표현된 데이터 수집의 요구 사항으로, 상기 쿼리를 수행할 기하학적 타겟 영역(geometric target space)을 식별하기 위한 공간 연산자(spatial operator)가 정의되는 것을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치를 제공한다.

발명의 효과

[18] 본 발명의 실시예에 따르면, 센서 네트워크 데이터베이스(SNDB)가 다양한 공간 연산자(spatial operator)를 처리하여 타겟 영역(target space)을 정교하게 표현할 수 있도록 기존 센서 네트워크 쿼리 언어(SNQL)를 확장할 수 있다. 따라서, 쿼리를 위한 영역 정보를 용이하게 표현하고 처리할 수 있고 효율적으로 쿼리를 처리할 수 있다.

[19] 본 발명의 실시예에 따르면, 수집된 중간 쿼리 결과를 기준으로 다시 쿼리를 수행하는 경우 쿼리를 쿼리의 중간 결과를 서버까지 전송하지 않고 노드 레벨인 LCA(lowest common ancestor) 노드에서 쿼리의 중간 결과를 이용하여 처리함으로써 쿼리 처리 비용을 줄일 수 있다.

[20] 본 발명의 실시예에 따르면, 다각형 모양의 공간(polygon shaped space)과 공간 연산자를 정의함으로써 센서 네트워크 데이터베이스(SNDB)에서 특정 영역 대상의 쿼리 처리 시 비정형 영역을 대상으로 쿼리 처리를 수행할 수 있다.

도면의 간단한 설명

[21] 도 1은 본 발명의 일 실시예에 있어서, 확장된 센서 네트워크 쿼리 언어(SNQL)의 엔벨롭(envelop)을 설명하기 위한 예시 도면이다.

[22] 도 2 내지 도 9는 본 발명의 일 실시예에 있어서, 확장된 센서 네트워크 쿼리 언어(SNQL)의 공간 연산자(spatial operator)를 실행하여 엔벨롭(envelop)을 형성하는 과정을 설명하기 위한 예시 도면이다.

[23] 도 10은 본 발명의 일 실시예에 있어서, LCA(lowest common ancestor) 노드를 이용한 쿼리 처리 과정을 설명하기 위한 예시 도면이다.

[24] 도 11은 본 발명의 일 실시예에 있어서, 확장된 센서 네트워크 쿼리

언어(SNQL)를 이용한 센서 네트워크 데이터베이스의 시스템 구조를 도시한 블록도이다.

- [25] 도 12는 본 발명의 일 실시예에 있어서, 메타데이터 관리를 설명하기 위한 예시 도면이다.

발명의 실시를 위한 최선의 형태

- [26] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

- [27] 본 실시예들은 무선 센서 네트워크에서 쿼리를 처리하는 기술에 관한 것으로, 특히 센서 네트워크 데이터베이스(SNDB)에서 에너지 효율성이 우수한 쿼리 처리 기술 및 비정형 영역의 타겟을 대상으로 하는 쿼리 처리 기술에 관한 것이다.

- [28] 센서 네트워크 쿼리 언어(SNQL)를 위한 공간 연산자

- [29] - 센서 네트워크 쿼리 언어(SNQL) 공간의 설계

- [30] 이하에서는 다양한 공간 연산을 지원하기 위한 센서 네트워크 쿼리 언어(SNQL)의 공간 연산자를 소개한다.

- [31] 센서 네트워크 쿼리 언어(SNQL)의 기하학적 타겟 영역은 공간적 쿼리 연산의 기본 단위인 SNQL 엔벨롭(envelope)이라 불리는 직사각형으로 표현된다. SNQL 엔벨롭의 개념은 GIS에 사용되는 ENVELOPE과 유사하기는 하나, 기하학적 오브젝트의 최소 한계 박스로 응답하는 GIS ENVELOPE과의 차이를 보이며, 본 발명에서의 SNQL 엔벨롭은 주어진 조건을 만족시키고 타겟 영역으로 응답하는 센서 노드의 그룹으로 식별된다.

- [32] SNQL 엔벨롭은 $\{\text{space-ID}, (x\_i, x\_j, y\_i, y\_j)\}$ 의 형태로 대표되는데, space-ID는 기하학적 공간 식별자이고, (x_i, x_j, y_i, y_j) 는 도 1에 도시된 바와 같이 $\{(x_i, y_i), (x_j, y_j), (x_i, y_j), (x_j, y_i)\}$ 인 엔벨롭의 꼭지점들을 의미하는 공간 표기이다.

- [33] 위치의 점 또한 $i=j$ 인 SNQL 엔벨롭의 일종으로 고려한다. 이는 1개의 센서 노드도 도 2(step1)에 도시된 바와 같이 엔벨롭을 형성한다는 것을 의미한다. 그러므로, 모든 SNQL 공간은 공간적 쿼리 연산을 위한 폐쇄 특성(closure property)을 성취하기 위한 SNQL 엔벨롭으로 표현된다.

- [34] 타겟 영역을 식별하는 SNQL 엔벨롭의 특성에 기초하여 유용한 공간 연산자를 설계할 수 있다.

- [35] 공간 식별 연산자(space Identification Operators)

- [36] - Envelope(Condition(s))

- [37] 엔벨롭 연산은 주어진 조건을 만족시키면서 기하학적 공간을 구성하는 센서들의 그룹으로 결과를 보일 것이다. 예를 들어, "온도가 30도를 초과한 센서 노드로 이루어진 공간을 찾아라"라는 쿼리는 $\text{Envelope}(\text{temp} > 30)$ 으로 표현될 수 있다.

- [38] SNQL 엔벨롭은 인접한 것들끼리 합병된다. 도 2는 리프 노드(leaf node)(가장 작은 엔벨롭)이 어떻게 더 큰 엔벨롭으로 합병되는지를 표현한 것이다. 도 2의

단계1(step1)에서, 노드1부터 노드8까지 8개의 노드가 SNQL 엔벨롭 e1부터 엔벨롭 e8까지를 형성하면서 쿼리 조건을 만족시킨다. 이후 단계2(step2)에서, SNQL 엔벨롭 e1, e2, e3는 e4, e5, e6과 합병되며, 상대적으로 새롭고 더 커진 SNQL 엔벨롭 e4, e5, e6를 생성한다. 이 과정은 단계3(step3)과 같은 새롭고 더 커진 SNQL 엔벨롭 e7이 형성될 때까지 반복된다. 반면에, 엔벨롭 e4는 더 이상 확장하지 않는데, 이는 노드9가 주어진 조건을 만족하지 않기 때문이다. 마침내, 엔벨롭 e4, e7과 e8은 Envelope(temp>30)의 결과물로서 베이스 스테이션으로 전송된다.

[39] 엔벨롭 연산의 과정은 표 1의 알고리즘과 같다.

[40]

[41] [표 1]

[42]

Function $Operator_{Envelope}$

Input: cond // condition(s)

Output: $S_{result} = \{e_m\}$ where $e_m = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$

Begin

(1) **If** N satisfy cond **then**

(2) create $e_N = \langle \text{timestamp}, (x_N, x_N, y_N, y_N) \rangle$

(3) **If** any envelopes created from N_{child} exist **then**

(4) merge envelopes created from N_{child} into e_N **End If**

(5) add e_N to S_{result} **End If**

(6) **return** S_{result}

End

[43]

[44] N 은 임의의 센서 노드를 의미한다. N 의 자식 노드는 N_{child} 로 정의되고, N 의 물리적 위치는 (x_N, y_N) 이다. 이 연산은 입력을 cond(예를 들어, 조건)와 같은 지표로 수신하고 기하적 공간의 집합을 $e_m = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$ 인 $S_{result} = \{e_m\}$ 의 형태로 리턴한다. 여기서 space-ID는 기하학적 공간 식별자이고, (x_i, x_j, y_i, y_j) 는 주어진 조건을 만족시키는 센서 노드 그룹을 식별하는 e_m 의 꼭지점을 의미한다. 우선, 만약 노드 N 이 조건을 만족한다면, SNQL 엔벨롭 e_N 은 $\langle \text{timestamp}, (x_N, x_N, y_N, y_N) \rangle$ 의 형태로 생성된다. 타임스탬프(Timestamp)는 엔벨롭의 고유성을 확보하기 위해 사용되고 (x_N, x_N, y_N, y_N) 은 (x_N, y_N) 에 위치한 노드 N 에 대한 엔벨롭 표기이다. 만약 다른 엔벨롭이 N_{child} 로부터 생성된다면, 이 엔벨롭은 더 큰 e_N 으로 합병된다. 그 결과, e_N 이 S_{result} 에 추가된다.

[45] - $NearBy(x_i, x_j, y_i, y_j)$

[46] $NearBy$ 연산은 특정된 목표 위치와 가장 가까운 노드의 기하학적 위치를 $i=j$ 인

{space-ID, (x_i, x_j, y_i, y_j) }의 형태로 생성하여 리턴한다. 쿼리는 노드들을 거쳐 타겟 위치까지 전송되고, 쿼리 경로에 있는 노드들은 쿼리를 등록한다.

[47] 도 3은 쿼리가 어떻게 타겟 위치에 가장 근접한 노드까지 전송되는 과정을 보여준다. 예를 들어, "좌표 $(x=10, y=20)$ 과 가장 가까운 센서 노드로 이루어진 공간을 찾으라."에 해당되는 쿼리는 $\text{NearBy}(10, 10, 20, 20)$ 으로 표현된다.

[48] 도 3의 단계1(step1)에서 알 수 있듯이, 주어진 좌표를 포함하는 서브트리에 해당되는 노드는 $\text{NearBy}(10, 10, 20, 20)$ 의 쿼리를 수신하고 등록한다. 그리고, 쿼리는 노드1부터 노드6까지를 따라 통과한다. 반면, 노드7부터 노드10은 노드5와 노드6이 그들의 서브트리가 주어진 좌표를 포함하지 않기 때문에 쿼리를 전송하지 않음에 따라 쿼리를 수신하지 않는다. 단계2(step2)에서, 노드4가 SNQL 엔벨롭 e_3 을 형성하고 노드5와 노드6에 의해 형성된 SNQL 엔벨롭 e_1 과 e_2 를 수신한다. 그리고 노드4는 SNQL 엔벨롭 e_1 을 주어진 타겟 위치로부터 가장 가까운 것으로 e_1 을 베이스 스테이션에 전송한다.

[49] NearBy 연산의 과정은 표2의 알고리즘과 같다.

[50]

[51] [표 2]

[52]

Function $\text{Operator}_{\text{NearBy}}$

Input: $\text{coord} = (x_i, x_j, y_i, y_j)$ //the specified coordinates of target location

Output: $S_{\text{result}} = \{e_m\}$ where $e_m = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$

Begin

(1) create $e_N = \langle \text{timestamp}, (x_N, x_N, y_N, y_N) \rangle$

(2) **If** any envelopes created from N_{child} exists **then**

(3) find the nearest envelope to coord among e_N and envelopes created from N_{child}

(4) set the nearest envelope to S_{result} **End If**

(5) **return** S_{result}

End

[53] NearBy 연산은 coord (예를 들어, 타겟 위치의 좌표)를 입력으로 수신하고 $e_m = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$ 인 $S_{\text{result}} = \{e_m\}$ 의 형태로 기하학적 공간의 집합을 리턴한다. 먼저, SNQL 엔벨롭 e_N 은 $\langle \text{timestamp}, (x_N, x_N, y_N, y_N) \rangle$ 의 형태로 생성된다. 만약, 다른 엔벨롭이 N_{child} 에 의해 생성되어 있다면, coord 와 가장 근접한 것을 찾기 위해 e_N 과 비교한다. 그리고, 가장 가까운 엔벨롭이 리턴된다.

[54] - $\text{Distance}(\text{Envelope}, \text{Range Value})$

[55] Distance 연산은 지정된 기하학적 공간으로부터 타 공간까지의 범위값을 식별한다. 상기한 바와 같이, 위치의 점 또한 SNQL 엔벨롭으로 표현한다는 것을 알고 있다. 예를 들어, "온도가 30도 이상인 기하학적 공간으로부터 10미터 떨어진 센서 노드로 이루어진 공간을 찾아라."라는 쿼리는 $\text{Distance}(\text{Envelope}(\text{temp} > 30), 10\text{m})$ 로 표현될 수 있다.

[56] 도 4의 단계1(step1)에서 알 수 있듯이, LCA 노드로서 노드5는 구성원이 온도 조건을 만족하는 두 SNQL 엔벨롭 e_1 과 e_2 를 식별한다. 단계2(step2)에서,

노드5는 주어진 범위 값에 따라 확장된 SNQL 엔벨롭 e_1 과 e_2 를 생성하고 확장된 SNQL 엔벨롭 e_1 과 e_2 에 쿼리를 전송한다.

[57] Distance 연산의 과정은 표 3의 알고리즘과 같다.

[58]

[59] <표 3>

[60]

Function $Operator_{Distance}$

Input: (1) $S_{in} = \{e_1, e_2, \dots, e_k\}$ where $e_k = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$

(2) ra // a range value

Output: $S_{result} = \{e'_1, e'_2, \dots, e'_k\}$ where e'_k is expanded envelope of e_k

Begin

(1) **For** each e_k in S_{in}

(2) Set envelope of e_k with $(x_i - ra, x_j + ra, y_i - ra, y_j + ra)$

(3) **EndFor**

(4) $S_{result} := S_{in}$

(5) **return** S_{result}

End

[61] Distance 연산의 함수는 기하학적 공간의 집합 $e_k = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$ 인 $S_{in} = \{e_1, e_2, \dots, e_k\}$ 와 범위 값을 입력으로 수신한다. e_k 가 기존 e_k 의 새롭게 확장된 엔벨롭인 $S_{result} = \{e'_1, e'_2, \dots, e'_k\}$ 를 리턴한다. 범위값에 따른 새로운 엔벨롭을 찾기 위해 각 e_k 의 envelope은 두 번째 입력 지표인 +/-의 범위값에 의해 확장된다. 그리고, S_{result} 가 리턴된다.

[62] - Direction(Envelope, Direction Value)

[63] Direction 연산은 지정된 공간 밖의 기하학적 공간을 동, 서, 남, 북의 주어진 방향과 이들의 혼합으로 표시한다. 예를 들어, "온도가 30도 이상인 어떤 공간의 서쪽에 위치한 센서 노드로 이루어진 공간을 찾아라."라는 쿼리는 Direction(Envelope(temp>30), WEST)로 표현된다.

[64] 도 5는 Direction 연산을 설명한다. 단계1(step1)에서, LCA 노드로서 노드5는 구성원이 온도조건을 만족하는 SNQL 엔벨롭 e_1 과 e_2 를 식별한다.

단계2(step2)에서, 노드5는 엔벨롭 e_2 를 엔벨롭 중 가장 서쪽에 있는 엔벨롭임을 인지하고 SNQL 엔벨롭 e_3 을 결과로 리턴한다.

[65] Direction 연산의 과정은 표 4의 알고리즘과 같다.

[66] Direction 연산의 함수는 기하학적 공간의 집합 $e_k = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$ 인 $S_{in} = \{e_1, e_2, \dots, e_k\}$ 와 방향값을 입력으로 수신한다. $e_m = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$ 인 $S_{result} = \{e_m\}$ 의 형태로 기하학적 공간의 집합을 결과로 리턴한다. S_{in} 으로 주어진 방향의 가장 끝 엔벨롭을 찾은 뒤, 주어진 방향 D 로 결과의 기하학적 엔벨롭 e_N 이 식별된다. 예를 들어, 방향이 북쪽으로 주어졌을 때, S_{in} 중 가장 북쪽의 엔벨롭이 찾아지고 엡실론 ϵ 으로 식별된다. 그리고, 공간좌표($X_{min}, X_{max}, \epsilon \cdot y_j, Y_{max}$)를 갖는 엔벨롭 e_N 을 $\epsilon \cdot y_j$ 보다 큰 y_i 를 갖는 결과공간으로서 생성한다. 여기서, X_{max} 나 X_{min}

은 영역의 x좌표의 최대값 또는 최소값을 의미하고, $Y_{\max}$ 나 $Y_{\min}$ 은 지역의 y좌표의 최대값 또는 최소값을 의미한다.

[67] <표 4>

[68]

Function *Operator*_{Direction}

Input: (1) $S_{\text{in}} = \{e_1, e_2, \dots, e_k\}$ where $e_k = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$
(2) D // a direction value

Output: $S_{\text{result}} = \{e_m\}$ where $e_m = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$

Begin

(1) **Switch**(D) // find geometric space of direction operation

(2) **Case** ($D = 90$) or ($D = \text{EAST}$):

(3) $\varepsilon := \text{easternmost } e_k \text{ from } S_{\text{in}}$

(4) create $e_N = \langle \text{timestamp}, (\varepsilon \cdot x_j, X_{\max}, Y_{\min}, Y_{\max}) \rangle$

(5) **Case** ($D = 270$) or ($D = \text{WEST}$):

(6) $\varepsilon := \text{westernmost } e_k \text{ from } S_{\text{in}}$

(7) create $e_N = \langle \text{timestamp}, (X_{\min}, \varepsilon \cdot x_i, Y_{\min}, Y_{\max}) \rangle$

(8) **Case** ($D = 0$) or ($D = 360$) or ($D = \text{NORTH}$)

(9) $\varepsilon := \text{northernmost } e_k \text{ from } S_{\text{in}}$

(10) create $e_N = \langle \text{timestamp}, (X_{\min}, X_{\max}, \varepsilon \cdot y_j, Y_{\max}) \rangle$

(11) **Case** ($D = 180$) or ($D = \text{SOUTH}$):

(12) $\varepsilon := \text{southernmost } e_k \text{ from } S_{\text{in}}$

(13) create $e_N = \langle \text{timestamp}, (X_{\min}, X_{\max}, Y_{\min}, \varepsilon \cdot y_i) \rangle$

(14) **End Switch**

(15) $S_{\text{result}} := e_N$

(16) **return** S_{result}

End

[69] 집합 연산자(Set Operators)

[70] 집합 연산자는 두 개의 피연산(operand) 공간을 사용함으로써 타겟 영역의 센서 노드를 식별하기 위해 설계된 것이다. 집합 연산자를 사용함으로써 SNQL은 응용프로그램으로 하여금 더 정교한 타겟 영역을 표현할 수 있게 한다. "온도가 30도 이상이면서 습도는 10%보다 낮고/낮지 않으며/낮거나 산의 서쪽 능선에 있는/있지 않은 복합 영역으로 식별되는 지역의 노드ID를 얻으라."라는 예시 쿼리가 있다. 이 쿼리에는 새로운 공간의 센서 노드를 식별하기 위해 Intersection, Union, Difference 연산을 사용한다.

[71] Union 연산과 Difference 연산은 결과의 최선의 정확도를 위해 도 6과 같이 다수의 분리된 공간을 생성한다는 것을 알고 있다. 주어진 공간은 다음과 같이 분리된 공간으로 분할된다. 단계1(step1)에서, $S_a = \{e_{a1}, e_{a2}\}$, $S_b = \{e_{b1}, e_{b2}\}$ 과 같은 두 개의 피연산 공간이 주어진다. 단계2(step2)에서, S_a 와 S_b 의 각 요소간 외적을 거쳐, intersect envelope e_{i1} e_{i2} 가 식별된다. 단계3(step3)에서, S_a S_b 를 구성하는 요소

envelope들은 그 공간이 동서남북, 그리고 그 조합에 의한 8방향으로 e_{i1} e_{i2} 를 둘러싸고 존재하는지 비교된다. 그리고 존재하는 공간이 $S_a = \{e_{s11}, e_{s12}, e_{s13}, e_{s23}, e_{s24}, e_{s25}\}$, $S_b = \{e_{s15}, e_{s16}, e_{s17}, e_{s21}, e_{s27}, e_{s28}\}$ 을 형성한다.

[72] - Intersection((s1, s2)|Envelope, Envelope)

[73] Intersection연산은 명확하게 space-ID로 특정되거나 두 개의 공간연산의 결과값으로 함축적으로 특정된(예를 들어, SNQL 엔벨롭) 두 피연산 공간을 사용한다. 예컨대, "온도가 30도 이상이고 습도는 10%이하인 산의 서쪽 능선에 있는 교차 지역의 노드 ID를 얻어라."라는 예시 쿼리를 이용한다. 이 쿼리는 주어진 조건을 만족하는 중첩되는 분리된 공간의 다수의 조합을 통해 실현 가능하고 그러므로 분리된 공간의 집합(예를 들어, SNQL 엔벨롭)이 결과로 돌아올 것이다.

[74] 도 7이 Intersection 연산의 실행을 보여준다. 단계1(step1)에서, LCA 노드로서 노드1이 온도가 30도 이상인 SNQL 엔벨롭 e1을 수신하고 습도가 10%이하인 SNQL 엔벨롭 e2를 수신한다. 단계2(step2)에서, 노드1이 엔벨롭 e1과 e2의 교차지역을 계산한다. 그리고, SNQL 엔벨롭 e3를 생성하고 e3에게 쿼리를 전송한다.

[75] - Union((s1, s2)|Envelope, Envelope)

[76] Union연산은 명확하게 space-ID로 특정되거나 두 개의 공간연산의 결과값으로 함축적으로 특정된(예를 들어, SNQL 엔벨롭) 두 피연산 공간을 사용한다. "온도가 30도 이상이고 습도는 10%이하인 산의 서쪽 능선에 있는 합집합 지역의 노드 ID를 얻어라."라는 예시 쿼리가 사용된다. 이 예시는 주어진 조건을 만족하는 중첩되는 분리된 공간의 다수의 조합을 통해 실현 가능함을 암시하고, 결과적으로 구성하는 분리된 공간의 집합이 결과로 돌아올 것이다(예를 들어, SNQL 엔벨롭).

[77] 도 8이 Union연산의 예시 실행을 나타낸다. 단계1(step1)에서, LCA노드로서 노드1이 온도가 30도 이상인 SNQL 엔벨롭 e1을 수신하고 습도가 10%이하인 SNQL 엔벨롭 e2를 수신한다. 단계2(step2)에서, 노드1은 엔벨롭 e1과 e2을 교차지역에 기초하여 분리함으로써 이들의 합집합 지역을 계산한다. 그리고 노드1은 엔벨롭 e3부터 e9까지를 생성하고 쿼리를 e3부터 e9까지의 센서 노드로 전송한다. Intersection연산과 달리 Union연산은 도 8과 같이 합쳐졌을 때 두 공간의 합집합 지역을 나타내는 엔벨롭의 집합을 생성한다는 것이다.

[78] - Difference((s1,s2)|Envelope, Envelope)

[79] Difference연산은 명확하게 space-ID로 특정되거나 두 개의 공간연산의 결과값으로 함축적으로 특정된(예를 들어, SNQL 엔벨롭) 두 피연산 공간을 사용한다. "온도가 30도 이상이고 습도는 10%이상인 산의 서쪽 능선에 있는 지역의 노드 ID를 얻어라."라는 예시 쿼리가 사용된다. 이 예시는 주어진 조건을 만족하는 중첩되는 분리된 공간의 다수의 조합을 통해 실현 가능함을 암시하고, 결과적으로, 구성하는 분리된 공간의 집합이 결과로 돌아올 것이다(예를 들어,

SNQL 엔벨롭).

- [80] 도 9가 Difference연산의 예시 실행을 보여준다. 단계1(step1)에서, LCA노드로서 노드1은 온도가 30도 이상인 SNQL 엔벨롭 e_1 을 수신하고 습도가 10%이하인 SNQL 엔벨롭 e_2 를 수신한다. 단계2(step2)에서, 노드1은 공통지역을 분할함으로써, e_1 과 e_2 의 차이가 나는 지역을 계산한다. 그리고, SNQL 엔벨롭 e_3, e_4, e_5 를 생성하고 쿼리를 e_3, e_4, e_5 노드에 전송한다. Union연산과 유사하게, Difference 연산은 도 9에 나타나듯 합쳐져서 Difference연산의 결과 분리된 공간을 형성하는 엔벨롭의 집합을 생성한다.

- [81] 상기한 Set연산의 과정은 표 5의 알고리즘과 같다.

- [82] <표 5>

- [83] **Function Operator**_{SetOperators}
Input: (1) op_name
 (2) $S_a = \{e_1, e_2, \dots, e_a\}$ where $e_a = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$
 (3) $S_b = \{e_1, e_2, \dots, e_b\}$ where $e_b = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$
Output: $S_{\text{result}} = \{e_1, e_2, \dots, e_n\}$ where $e_n = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$
Begin
 (1) **For** each e_a in S_a
 (2) **For** each e_b in S_b
 (3) **If** envelope of e_a overlap that of e_b **then**
 (4) create intersect envelope $e_i = \langle \text{timestamp}, (x_i, x_j, y_i, y_j) \rangle$
 (5) create $\varepsilon_a = \{e_1, e_2, \dots, e_k\}$ where e_k is constituent envelope of e_a by splitting e_a surrounding e_i
 (6) create $\varepsilon_b = \{e_1, e_2, \dots, e_k\}$ where e_k is constituent envelope of e_b by splitting e_b surrounding e_i
 (7) **If** (op_name = "Intersection") **then** add e_i to S_{result} **End If**
 (8) **If** (op_name = "Difference") **then** add ε_a to S_{result} **End If**
 (9) **If** (op_name = "Union") **then** add $\varepsilon_a, \varepsilon_b, e_i$ to S_{result} **End If**
 (10) **Else**
 (11) **If** (op_name = "Difference") **then** add e_a to S_{result} **End If**
 (12) **If** (op_name = "Union") **then** add e_a, e_b to S_{result} **End If**
 (13) **End If**
 (14) **EndFor**
 (15) **EndFor**
 (16) remove redundant envelopes from S_{result}
 (17) **return** S_{result}
End

- [84] Set연산의 함수는 op_name(예를 들면, 연산자 이름), $S_a = \{e_{s11}, e_{s12}, e_{s13}, e_{s23}, e_{s24}, e_{s25}\}$, $S_b = \{e_{s15}, e_{s16}, e_{s17}, e_{s21}, e_{s27}, e_{s28}\}$ 을 입력으로 수신하고 $e_m = \{\text{space-ID}, (x_i, x_j, y_i, y_j)\}$ 인 $S_{\text{result}} = \{e_m\}$ 의 형태로 기하학적 공간의 집합을 결과로 리턴한다(알고리즘의 1-4줄). 우선, S_a 내 각 엔벨롭 e_a 들이 S_b 의 e_b 와 겹칠 때, 교차 엔벨롭 e_i 가 $\langle \text{timestamp}, (x_N, x_N, y_N, y_N) \rangle$ 의 형태로 생성된다. 이를 8방향(동서남북 그리고 그 조합)과 비교하여, e_a 가 각 방향의 e_i 의 옆에 위치할 때, e_a 는 방향별 구성 엔벨롭으로 나뉜다. 그리고, 방향별 구성 엔벨롭들은 $\varepsilon_a = \{e_1, e_2, e_3, \dots, e_k\}$ 를 형성한다. 또한, e_b 도 같은 방식으로 나뉜다. 여기서, 엔벨롭은 e_i 에 위치에 의해 식별된다. 도 8의 경우에서, e_3 은 교차점 엔벨롭으로서 생성되고, $\varepsilon_a = \{e_7, e_8, e_9\}$, $\varepsilon_b = \{e_4, e_5, e_6\}$ 가 e_3 을 둘러싸고 생성되는 반면 e_3 의 북서와 동서쪽 방향으로 엔벨롭들이 생성되지 않는다(알고리즘의 5-6줄). 만약 op_named이 "Intersection"이라면, e_i 는 S_{result} 에

추가된다(알고리즘의 7줄). 만약 op_name 이 "Difference"라면, e_a 가 S_{result} 에
 추가된다(알고리즘의 8줄). 만약 op_name 이 "Union"이라면, e_a , e_b , e_i 가 S_{result} 에
 추가된다(알고리즘의 9줄). 만약 겹치는 부분이 존재하지 않는다면, op_name 이
 "Difference"일 때 e_a 가 S_{result} 에 추가되는 한편 op_name 이 "Union"일 때는 e_a 와 e_b 가
 S_{result} 에 추가된다(알고리즘의 10-12줄). 마침내 S_a 와 S_b 의 내적에 의해 발생한
 불필요한 엔벨롭은 S_{result} 로부터 제거된다(알고리즘의 16줄).

[85] SNQL에서의 공간 연산자는 표 6과 같다.

[86] <표 6>

[87]

Divisio	Operator	Description
Space identification operators	Envelope	Return the geometric spaces that satisfy a given condition
	NearBy	Return the geometric space that is the nearest to the specified point of location
	Distance	Return the geometric spaces that satisfy within the given distance from the given geometric spaces
	Direction	Return the geometric space that satisfies the given direction from the given geometric spaces
Set operators	Intersection	Return the geometric spaces that lie in the intersection area of the given geometric spaces
	Union	Return the geometric spaces that lie in the union area of the given geometric spaces
	Difference	Return the geometric spaces that lie in the difference area of the given geometric spaces

[88] 센서 네트워크 쿼리 언어(SNQL)에서의 공간적 표현

[89] 공간 연산자를 지원 할 수 있도록 센서 네트워크 쿼리 언어(SNQL)를 확장하였다. 기존 센서 네트워크 쿼리 언어(SNQL)는 Select-From-Where 구조에 기초하여 만들어졌고 일회성, 연속형 그리고 이벤트에 기반의 쿼리를 지원한다.

[90] 표 7의 알고리즘은 기존 센서 네트워크 쿼리 언어(SNQL)의 기본 구조를 나타낸 것이다.

[91]

[92] <표 7>

[93]

EVENT ON *event-type (paramlist)*
SELECT *selectlist*
FROM *fromlist*
WHERE *expression*
SAMPLE PERIOD *time [FOR time]*
OUTPUT ACTION *command*
WITHIN *percentage*
CASE WHEN *event | expression*
 THEN *Action | Sample Period | Within*
END;

[94]

[95] **EVENT ON** 항목은 이벤트가 발생했을 때, 다음 쿼리가 실행되도록 미리 설정된 이벤트이다. **SELECT** 항목은 각각의 센서 노드로부터 모아진 센서 데이터에 상응하는 속성을 묘사한다. **FROM** 항목은 쿼리 전달을 위한 초기 타겟 영역을 의미한다. **WHERE** 항목은 selection 조건을 묘사한다. **SAMPLE PERIOD** 항목은 데이터 수집을 위한 다양한 시간에 대한 표시이다. **OUTPUT ACTION** 항목은 이벤트나 선택조건이 만족되었을 때 실행될 센서의 액션을 정의한다. **WHITHIN** 항목은 모든 센서 노드 중 쿼리 과정에 참가하는 센서 노드의 비율을 의미한다. **CASE WHEN** 항목은 조건에 따른 쿼리의 브랜칭을 지정한다.

[96] 센서 네트워크 쿼리 언어(SNQL)에 공간 연산자를 추가하기 위하여, **WHERE** 항목과 **WHEN** 항목의 표현을 표 8처럼 확장하였다.

[97]

[98] <표 8>

[99]

```

<Expression> ::= <SQLAndExp> ("OR" <SQLAndExp>)
<SQLAndExp> ::= <SQLRelationalExp> ("AND" <SQLRelationalExp>)
<SQLRelationalExp> ::= <SQLSimpleExp> | <SQLExpressionList> | <SQLInClause> | <SQLBetweenClause>
<SQLExpressionList> ::= <SQLSimpleExp> ("," <SQLSimpleExp>)
<SQLSimpleExp> ::= <AggregateFunc> | <Count> | <NUMBER> | <STRING> | <BIND> | <SpatialFunc>
<SpatialFunc> ::= <Envelope> | <Nearby> | <Intersection> | <Union> | <Difference> | <Distance> | <Direction>
<Envelope> ::= "ENVELOPE (" <attribute> <operator> <constant> ")"
<NearBy> ::= "NEARBY (" <envelope> ")"
<Intersection> ::= "INTERSECTION (" <SpatialFunc> "," <SpatialFunc> ")"
<Union> ::= "UNION (" <SpatialFunc> "," <SpatialFunc> ")"
<Difference> ::= "DIFFERENCE (" <SpatialFunc> "," <SpatialFunc> ")"
<Distance> ::= "DISTNACE (" <SpatialFunc> "," <constant> ")"
<Direction> ::= "DIRECTION (" <SpatialFunc> "," <constant> ")"
<envelope> ::= "(" <integer> "," <integer> "," <integer> "," <integer> ")"
<attribute> ::= <string>
<operator> ::= ">" | "<" | "=" | "<>" | ">=" | "<="
<constant> ::= <integer> | <float> | <string>

```

- [100] 더 구체적으로는, 앞서 설명한 공간 연산자를 사용하기 위해서 <SpatialFunc>를 <Expression>의 <SQLSimpleExp>에 추가하였다. <SpatialFunc>는 <Envelope>, <NearBy>, <Distance>, <Direction>, <Intersection>, <Union>, <Difference>로 구성되어 있다. <Envelope>은 속성(예를 들어 온도), 연산자(예를 들어 ">", "<", "=",)와 상수가 혼합된 형태로 표현된다. <NearBy>는 x와 y좌표를 표시하는 정수값들로 표현된다. <Distance>는 <SpatialFunc>로부터 선택된 연산자와 거리를 가리키는 상수로 표현된다. <Direction>은 <SpatialFunc>로부터 선택된 연산자와 방향을 가리키는 상수로 표현된다. <Intersection>은 교차영역을 구하기 위한 <SpatialFunc>의 조합으로 구성되어 있다. <Union>은 합집합영역을 구하기 위한 <SpatialFunc>의 조합으로 구성되어 있다. <Difference>는 차집합영역을 구하기 위한 <SpatialFunc>의 조합으로 구성되어 있다. <envelope>, <attribute>, <operator>, <constant>는 기초 속성들로 추가적인 설명은 생략한다.
- [101] LCA: In-Network Query Processing Algorithm
- [102] 센서 네트워크 쿼리는 단일 쿼리내에서 다중 데이터 리딩이 자주 일어난다. 예를 들어, "습도가 10%이하인 산의 서쪽능선 지역의 온도를 얻어라"라고 주어진 예제 쿼리는 두 개의 분리된 태스크를 포함한다: (1)서부능선의 모든 노드로부터 습도를 읽어오는 것, (2)습도가 10%이하인 노드의 온도를 읽어오는 것이 그 것이다. 현존하는 시스템에서 이 작업은 베이스스테이션과 지정된 공간의 타겟 노드 간에 두 번의 순회하는 질의를 필요로 한다.
- [103] 본 발명에 따른 쿼리 처리 알고리즘은 다중의 쿼리 태스크가 LCA라 불리는 인-네트워크 노드에 의해 처리되는 것이 가능하게 한다. LCA노드는 타겟 영역의 모든 자식 노드를 모두 포함하도록 최소한 그리고 최초로 쿼리를 전송하는 중간 조상 노드의 일종이다. 이는 다중 리딩 태스크를 위해 인-네트워크 쿼리의 재구성 및 재전송의 역할을 하기 위해 설계되는 것이다.
- [104] 본 발명에 따른 쿼리 처리 알고리즘은 베이스스테이션(예를 들어, 루트 노드)과 타겟 노드 간 불필요한 다중의 쿼리/데이터 교환에 의해 발생하는 무선 센서 노드의 에너지 소비를 효율적으로 줄여준다.
- [105] LCA노드를 찾는 과정은 다음과 같다.
- [106] 쿼리 경로에 있는 각 노드는 두 지표를 관리한다(#Ancestor_{max_send}, #Child_{send}). 전자는 조상 노드가 자식 노드로 쿼리를 전송한 최대 개수를 의미하고, 후자는 쿼리를 전달한 자식 노드의 숫자를 의미한다. LCA노드는 #Child_{send} 값이 쿼리 경로 중 첫 번째로 1보다 커지는노드로 식별된다.
- [107] 도 10 (a)는 LCA노드가 어떻게 예시 쿼리 첫 타겟 영역에서 결정되는지를 보여준다. 베이스스테이션으로서 N01은 쿼리를 수신하고 쿼리 프로세서는 타겟 영역의 수신 노드인 N02를 찾는다. 여기서, N01은 조상 노드를 가지고 있지 않기 때문에, #Ancestor_{max_send}가 0이다. 그리고, 쿼리를 N02로만 보내기 때문에, #Child_{send}가 1이다. 이러한 방식으로, 첫 목표까지의 쿼리경로에 있는 N03, N04, N05, N06은 (#Ancestor_{max_send}=1, #Child_{send}=1)을 갖는다. N07은 N06으로부터 쿼리를

수신하고 두 개의 자식 노드로 쿼리를 전달한다. 그래서 처음으로 자식 노드의 수가 1보다 크게 된다($\#Ancestor_{max\_send}=1$, $\#Child_{send}=2$). 그러므로, N07은 이 쿼리의 타겟 영역의 모든 노드를 최소로 커버하는 최저의 중간 노드인 LCA노드가 된다.

[108] 도 10 (b)에서, LCA노드는 최종 목표 노드에서(예를 들어 최초의 타겟 영역에서 습도가 10%이하인 노드) 데이터 수집을 위한 쿼리를 다시 생성한다. 이때, LCA노드는 최종 목표 노드를 식별하는 데 중요한 역할을 하고 다음 쿼리 업무를 그 노드에 재전송한다(예를 들어 온도 수집).

[109] LCA 연산의 과정은 표 9의 알고리즘과 같다.

[110] LCA 연산의 함수는 $\#Ancestor_{max\_send}$ 를 입력으로 수신하고 현재 노드가 LCA노드인지 아닌지에 대한 결정에 따라 참/거짓을 결과로 리턴한다. 우선, 등록된 쿼리는 자식 노드로부터 수신된 엔벨롭으로 재구성된다. 만약 현재 노드가 다음의 조건을 만족한다면: (1) 노드는 ($\#Ancestor_{max\_send}=1$, $\#Child_{send}>1$), (2) 베이스스테이션은 ($\#Ancestor_{max\_send}=0$, $\#Child_{send}>1$)(알고리즘의 1-2줄). 그리고, 쿼리 프로세서(또는 Disseminator)은 재구성된 쿼리를 재전송하기 위해 불려진다. 그리고 참이 반환된다(알고리즘의 3-5줄). 만약, 현재 노드가 LCA노드가 아니라면, 거짓이 반환된다(알고리즘의 6줄).

[111] <표 9>

[112] **Function LCA**

Input: (1) $\#Ancestor_{max\_send}$
(2) $\#Child_{send}$

Output: a Boolean value // true or false

Begin

(1) **If** ($(\#Ancestor_{max\_send} = 1 \text{ and } \#Child_{send} > 1)$

|| $(\#Ancestor_{max\_send} = 0 \text{ and } \#Child_{send} > 1)$) **then**

(2) reform the registered query with envelopes receive from child nodes

(3) **If** the current node is base-station **then** call the Planner of Query Processor

(4) **Else** call the Disseminator **End If**

(5) **return true**

(6) **Else return false End If**

End

[113] Architecture

[114] 본 발명에 따른 센서 네트워크 데이터베이스 시스템의 구조를 설명한다.

[115] - 개요

[116] 도 11을 참조하면, 본 발명에 따른 센서 네트워크 데이터베이스 시스템은 SNQL 서버 역할을 하는 베이스스테이션(110), 및 타겟 영역 내에 위치한 센서 노드(120)로 구성되어 있다.

[117] 베이스스테이션(110)에서 쿼리를 받는 중, 쿼리 프로세서(111)는 쿼리를 분석하고 실행을 계획한 후 분석된 쿼리를 네트워크 매니저(Network

Manager)(112)에게 전송한다. 이는 노드와 노드 간 연결(예를 들어 토폴로지)에 대한 메타데이터를 통해 목표 노드까지의 경로를 결정하는 데에 사용한다. 최초의 타겟 영역의 센서 노드(120)까지 기록된 경로에 있는 노드들은 분석된 쿼리를 목표 노드의 조상 노드로 등록하고, 노드들 중 하나가 다중의 공간 연산이 적용될 때 새로운 타겟 노드를 식별하기 위한 LCA노드의 역할을 한다.

- [118] 타겟 영역의 노드인 센서 노드(120)의 쿼리 프로세서(121)는 실행자(Executor), 이벤트 매니저(Event Manager), 액션 매니저(Action Manager), 및 전송자(Disseminator)를 포함한다.
- [119] 센서 노드(120)에서 실행자는 쿼리를 실행하고 조상 노드를 통해 SNQL 서버로 데이터를 전송하고, 이벤트 매니저는 타겟 노드에 등록된 이벤트 쿼리를 관리한다. 그리고, 액션 매니저는 쿼리 조건이 만족되었을 때 실행되어야 할 액션(예를 들어 경보음 발생 등)을 관리하고, 전송자는 재구성된 쿼리를 다른 노드로 재전송 하는 역할을 한다.
- [120] 네트워크 매니저(122)는 목표 지역의 노드들까지 분석된 쿼리를 전달하고, 쿼리 결과를 수신하며, 자신의 조상 노드로 결과를 리턴한다. 결국, 베이스스테이션(110)은 어플리케이션으로 전달하기 전 주어진 쿼리에 의해 요청된 형식으로 수집된 센서 데이터를 후처리한다.
- [121] - 메타데이터 관리
- [122] 속성종류, 위치 정보, 센서노드의 MBR(minimum boundary rectangle)과 같은 SNQL의 메타데이터는 쿼리 처리과정 중 사용된다. 각 노드는 직접 연결된 자식 노드의 메타데이터를 다음과 같은 형식으로 관리한다: $\{(x_i, y_i), (NodeID, \{sensorType\})\}$ 의 형식인데, (x_i, y_i) 는 자식 노드의 좌표를 의미하고, $(NodeID, \{sensorType\})$ 는 nodeID와 센서종류를 의미한다. 또한, 서브트리의 메타데이터는 다음과 같은 형식으로 표현된다: $\{(x_i, x_j, y_i, y_j), \{sensorType\}\}$ 의 형태인데, (x_i, x_j, y_i, y_j) 는 MBR의 꼭짓점을 의미하고 $\{sensorType\}$ 는 서브트리의 센서 종류를 의미한다.
- [123] 도 12는 메타데이터 관리의 예시를 보여준다. 원은 센서 노드를 의미하고, 점선은 그 서브트리의 MBR을 의미한다. 베이스스테이션은 " $\{(10, 60), (N01, \{sensorType\})\}$ "을 직접 연결된 자식 노드의 메타데이터로 관리하고 " $\{(10, 50, 40, 100), \{sensorType\}\}$ "을 서브트리의 메타데이터로 관리한다. 또한 N01노드는 " $((25,95),(N02,\{sensorType\})), ((22,58),(N03,\{sensorType\}))$ "을 직접 연결된 자식 노드의 메타데이터로 관리하고 " $((25, 45, 77,100),\{sensorType\}), ((22,50,40,58,\{sensorType\}))$ "을 서브트리의 메타데이터로 관리한다. 이러한 방식으로, N02는 두 개의 직접 연결된 자식노드의 메타데이터를 관리하고 MBR을 포함한 두 개의 서브트리의 메타데이터를 관리한다. 여기서, 서브트리의 MBR은 타겟 영역이 서브트리의 MBR과 겹칠 때 자식 노드로 쿼리를 전달할 것인지 결정하는 데에 사용한다. 이 메타데이터는 센서 네트워크가 구성될 때 생성된다. 또한, 노드가 추가되거나 삭제될 때, 메타데이터는 그 조상 노드와

자식 노드를 업데이트 함으로서 재구성된다.

- [124] 이와 같이, 본 발명의 실시예에 따르면, 센서 네트워크 데이터베이스(SNDB)가 다양한 공간 연산자(spatial operator)를 처리하여 타겟 영역(target space)을 정교하게 표현할 수 있도록 기존 센서 네트워크 쿼리 언어(SNQL)를 확장하였다. 따라서, 쿼리를 위한 영역 정보를 용이하게 표현하고 처리할 수 있고 효율적으로 쿼리를 처리할 수 있다. 더욱이, 본 발명의 실시예에 따르면, 수집된 중간 쿼리 결과를 기준으로 다시 쿼리를 수행하는 경우 쿼리의 중간 결과를 서버까지 전송하지 않고 노드 레벨인 LCA(lowest common ancestor) 노드에서 쿼리의 중간 결과를 이용하여 처리함으로써 쿼리 처리 비용을 줄일 수 있다. 그리고, 본 발명의 실시예에 따르면, 다각형 모양의 공간(polygon shaped space)과 공간 연산자를 정의함으로써 센서 네트워크 데이터베이스(SNDB)에서 특정 영역 대상의 쿼리 처리 시 비정형 영역을 대상으로 쿼리 처리를 수행할 수 있다.
- [125] 본 발명의 실시예에 따른 방법들은 다양한 컴퓨터 시스템을 통하여 수행될 수 있는 프로그램 명령(instruction) 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다.
- [126] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [127] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상

장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

- [128] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

발명의 실시를 위한 형태

- [129] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.
- [130] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.
- [131]
- [132]

청구범위

- [청구항 1] 컴퓨터로 구현되는 무선 센서 네트워크에서의 쿼리 처리 방법에 있어서, 상기 무선 센서 네트워크에 속하는 기지국에서 상기 무선 센서 네트워크 상에 쿼리를 전송하는 단계; 및
상기 쿼리를 수신한 센서 노드에서, 상기 쿼리를 수행함에 따라 센서 데이터를 수집하는 단계
를 포함하고,
상기 쿼리는 센서 네트워크 쿼리 언어(sensor network query language, SNQL)로 표현된 데이터 수집의 요구 사항으로, 상기 쿼리를 수행할 기하학적 타겟 영역(geometric target space)을 식별하기 위한 공간 연산자(spatial operator)가 정의되는 것
을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.
- [청구항 2] 제1항에 있어서,
상기 공간 연산자는 주어진 조건(condition)을 만족시키는 기하학적 공간을 타겟 영역으로 설정하기 위한 엔벨롭 연산자(envelope operator)를 포함하는 것
을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.
- [청구항 3] 제2항에 있어서,
상기 공간 연산자는 타겟 위치와 가장 가까운 기하학적 공간을 타겟 영역으로 설정하기 위한 니어바이 연산자(nearby operator), 주어진 기하학적 공간으로부터 주어진 거리 이내에 있는 공간을 타겟 영역으로 설정하기 위한 거리 연산자(distance operator), 주어진 기하학적 공간으로부터 주어진 방향에 있는 공간을 타겟 영역으로 설정하기 위한 방향 연산자(direction operator) 중 적어도 하나의 연산자를 더 포함하는 것
을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.
- [청구항 4] 제2항에 있어서,
상기 공간 연산자는 둘 이상의 피연산자(operand)를 사용하는 집합 연산자(set operator)로서, 주어진 기하학적 공간 간의 교차 지역(intersection area)을 타겟 영역으로 설정하기 위한 교차 연산자(intersection operator), 주어진 기하학적 공간 간의 합집합 지역(union area)을 타겟 영역으로 설정하기 위한 합집합 연산자(union operator), 주어진 기하학적 공간 간의 차집합 지역(difference area)을 타겟 영역으로 설정하기 위한 차집합 연산자(difference operator) 중 적어도 하나의 연산자를 포함하는 것
을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.
- [청구항 5] 제1항에 있어서,
상기 센서 데이터를 수집하는 단계는,

상기 쿼리가 둘 이상의 태스크를 포함하는 경우,
 상기 쿼리를 전달하는 자식 노드의 수를 확인하는 단계;
 상기 자식 노드의 수에 따라 LCA(lowest common ancestor) 노드 여부를 결정하는 단계;
 상기 LCA 노드에 해당되는 경우 상기 자식 노드로부터 수신된 센서 데이터를 이용하여 상기 쿼리를 재구성하는 단계; 및
 상기 재구성된 쿼리를 상기 자식 노드로 재전송하는 단계를 포함하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.

[청구항 6] 제5항에 있어서,
 상기 LCA 노드 여부를 결정하는 단계는,
 상기 쿼리의 전송 경로에서 최초로 쿼리가 전달된 상기 자식 노드의 수가 1을 초과하는 경우 상기 LCA 노드로 결정하는 것을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.

[청구항 7] 제1항에 있어서,
 상기 센서 노드는 직접 연결된 적어도 하나의 자식 노드와 서브트리(subtree)의 메타데이터를 관리하고,
 상기 자식 노드의 메타데이터는 상기 자식 노드의 좌표와 식별 번호 및 센서 종류를 포함하고,
 상기 서브트리의 메타 데이터는 상기 서브트리의 MBR(minimum boundary rectangle)의 좌표와 센서 종류를 포함하는 것을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.

[청구항 8] 제7항에 있어서,
 상기 서브트리의 MBR이 상기 타겟 영역에 해당되는 경우 상기 서브트리에 포함되는 자식 노드로 상기 쿼리가 전달되는 것을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 방법.

[청구항 9] 적어도 하나의 프로그램이 로딩된 메모리; 및
 적어도 하나의 프로세서
 를 포함하고,
 상기 적어도 하나의 프로세서는, 상기 프로그램의 제어에 따라,
 무선 센서 네트워크에 속하는 기지국에서 상기 무선 센서 네트워크 상에 쿼리를 전송하는 과정; 및
 상기 쿼리를 수신한 센서 노드에서, 상기 쿼리를 수행함에 따라 센서 데이터를 수집하는 과정
 을 처리하고,
 상기 쿼리는 센서 네트워크 쿼리 언어(sensor network query language, SNQL)로 표현된 데이터 수집의 요구 사항으로, 상기 쿼리를 수행할 기하학적 타겟 영역(geometric target space)을 식별하기 위한 공간 연산자(spatial operator)가 정의되는 것

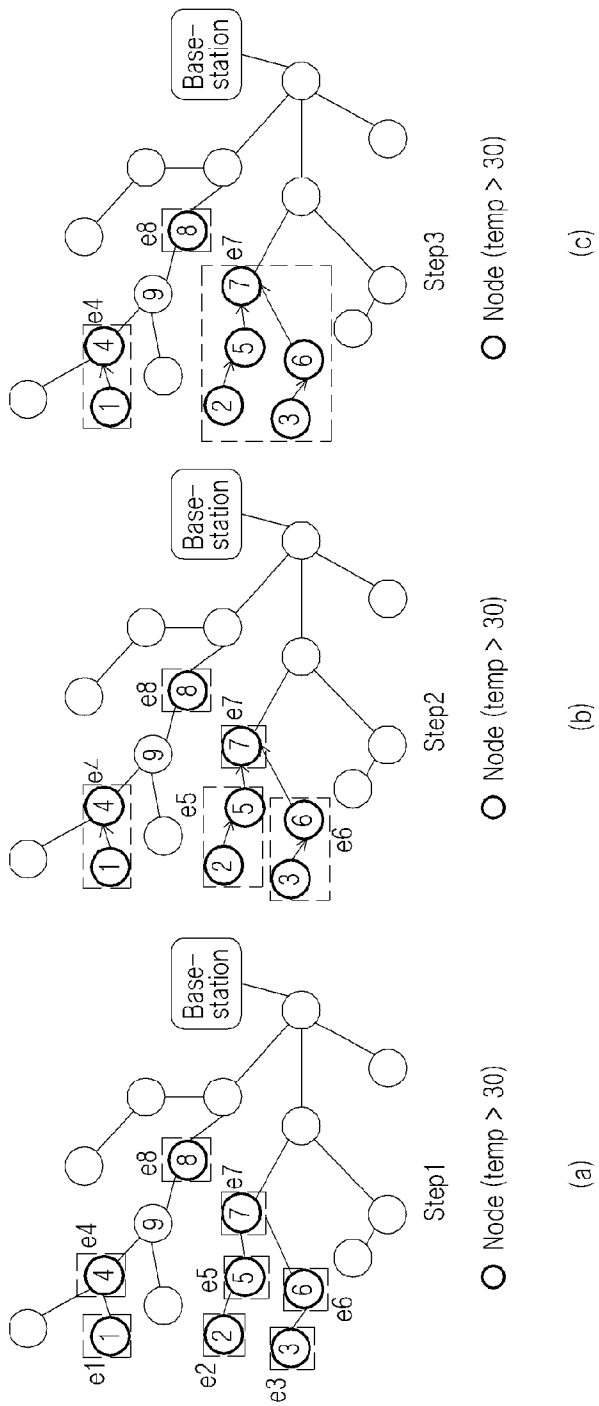
- 을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.
- [청구항 10] 제9항에 있어서,
상기 공간 연산자는 주어진 조건(condition)을 만족시키는 기하학적 공간을 타겟 영역으로 설정하기 위한 엔벨롭 연산자(envelope operator)를 포함하는 것
- 을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.
- [청구항 11] 제10항에 있어서,
상기 공간 연산자는 타겟 위치와 가장 가까운 기하학적 공간을 타겟 영역으로 설정하기 위한 니어바이 연산자(nearby operator), 주어진 기하학적 공간으로부터 주어진 거리 이내에 있는 공간을 타겟 영역으로 설정하기 위한 거리 연산자(distance operator), 주어진 기하학적 공간으로부터 주어진 방향에 있는 공간을 타겟 영역으로 설정하기 위한 방향 연산자(direction operator) 중 적어도 하나의 연산자를 더 포함하는 것
- 을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.
- [청구항 12] 제10항에 있어서,
상기 공간 연산자는 둘 이상의 피연산자(operand)를 사용하는 집합 연산자(set operator)로서, 주어진 기하학적 공간 간의 교차 지역(intersection area)을 타겟 영역으로 설정하기 위한 교차 연산자(intersection operator), 주어진 기하학적 공간 간의 합집합 지역(union area)을 타겟 영역으로 설정하기 위한 합집합 연산자(union operator), 주어진 기하학적 공간 간의 차집합 지역(difference area)을 타겟 영역으로 설정하기 위한 차집합 연산자(difference operator) 중 적어도 하나의 연산자를 포함하는 것
- 을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.
- [청구항 13] 제9항에 있어서,
상기 센서 데이터를 수집하는 과정은,
상기 쿼리가 둘 이상의 태스크를 포함하는 경우,
상기 쿼리를 전달하는 자식 노드의 수를 확인하는 과정;
상기 자식 노드의 수에 따라 LCA(lowest common ancestor) 노드 여부를 결정하는 과정;
상기 LCA 노드에 해당되는 경우 상기 자식 노드로부터 수신된 센서 데이터를 이용하여 상기 쿼리를 재구성하는 과정; 및
상기 재구성된 쿼리를 상기 자식 노드로 재전송하는 과정
- 을 포함하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.
- [청구항 14] 제13항에 있어서,
상기 LCA 노드 여부를 결정하는 과정은,
상기 쿼리의 전송 경로에서 최초로 쿼리가 전달된 상기 자식 노드의 수가 1을 초과하는 경우 상기 LCA 노드로 결정하는 것

- [청구항 15] 을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.
제9항에 있어서,
상기 센서 노드는 직접 연결된 적어도 하나의 자식 노드와 서브트리(subtree)의 메타데이터를 관리하고,
상기 자식 노드의 메타데이터는 상기 자식 노드의 좌표와 식별 번호 및 센서 종류를 포함하고,
상기 서브트리의 메타 데이터는 상기 서브트리의 MBR(minimum boundary rectangle)의 좌표와 센서 종류를 포함하는 것
을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.
- [청구항 16] 제7항에 있어서,
상기 서브트리의 MBR이 상기 타겟 영역에 해당되는 경우 상기 서브트리에 포함되는 자식 노드로 상기 쿼리가 전달되는 것
을 특징으로 하는 센서 네트워크 데이터베이스에서의 쿼리 처리 장치.

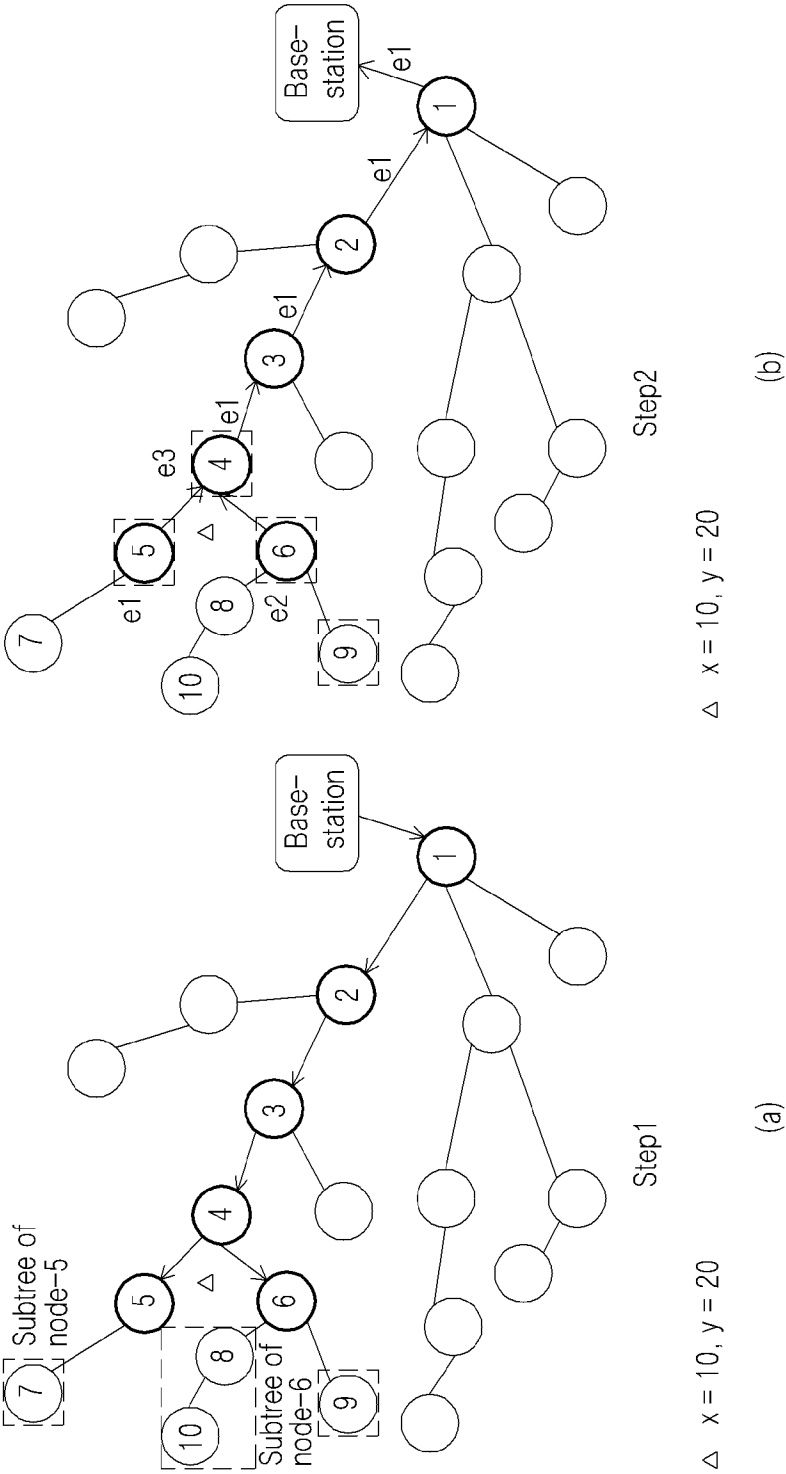
[도1]



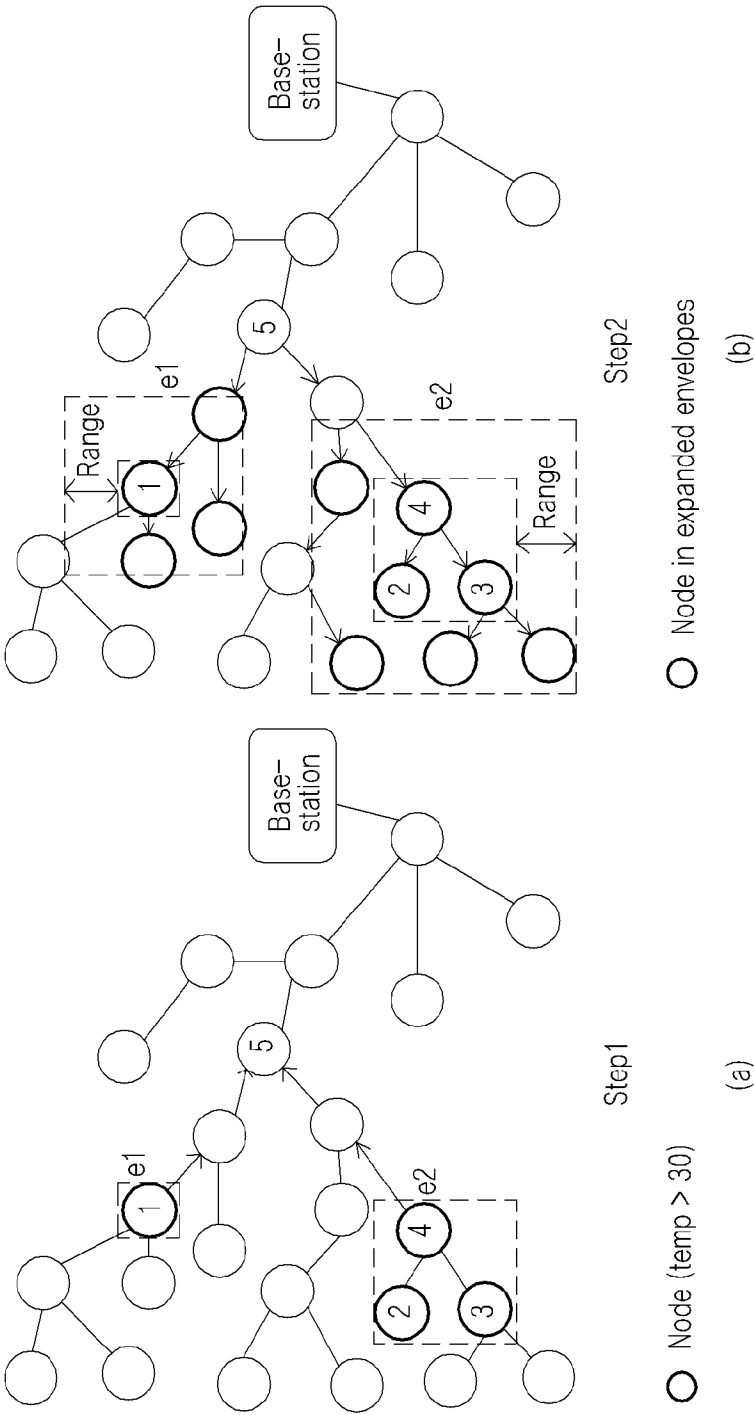
[도2]



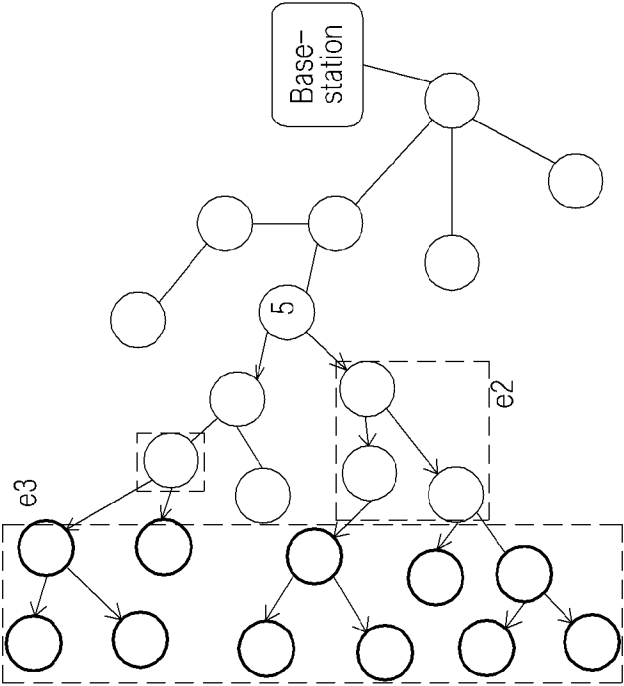
[도3]



[도4]



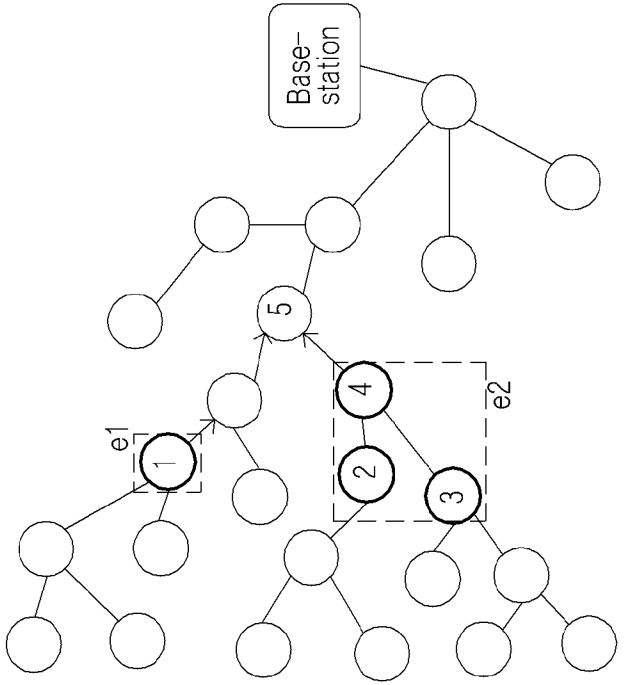
[도5]



Step2

○ Nodes in the west from the base envelope (e2)

(b)

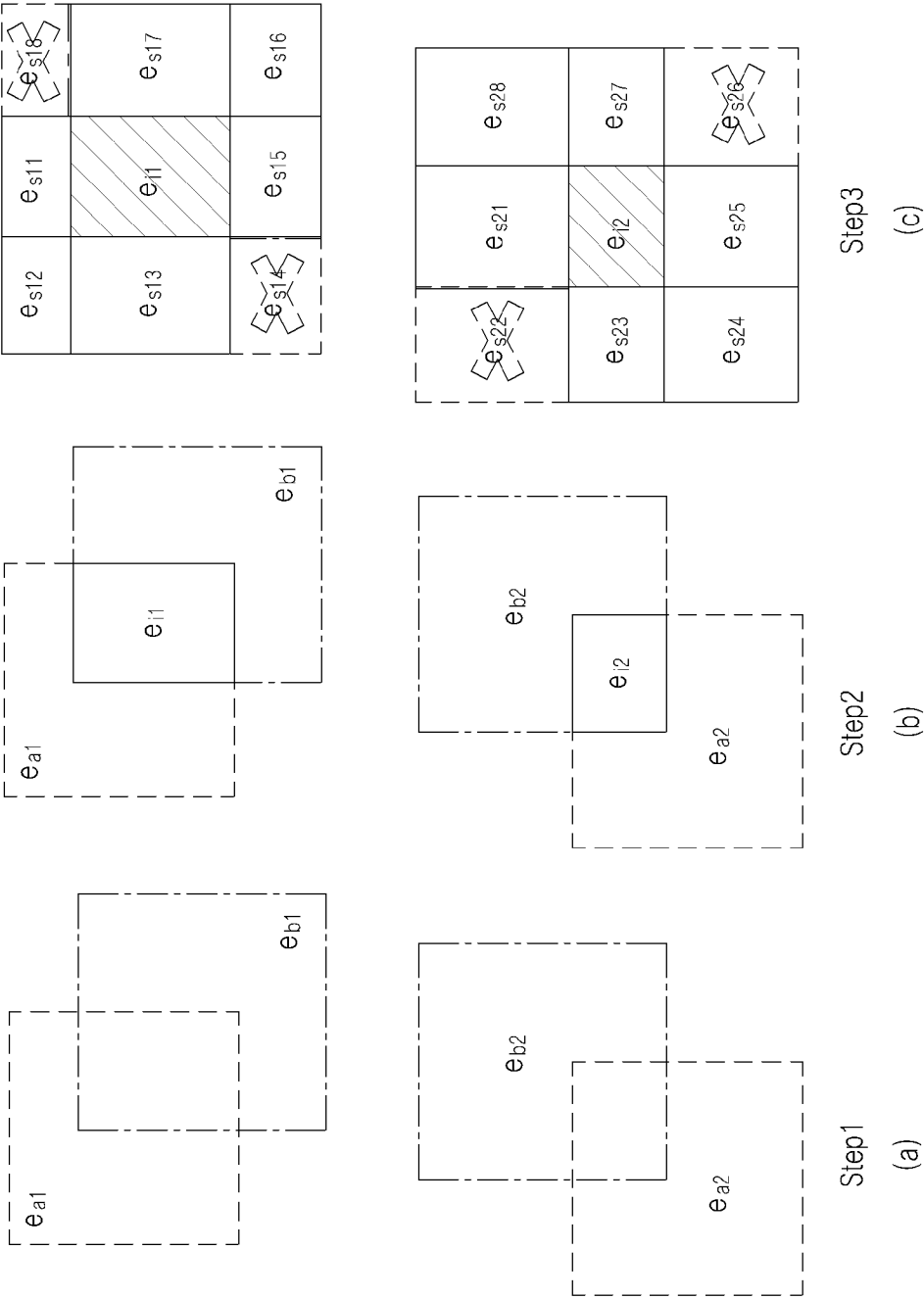


Step1

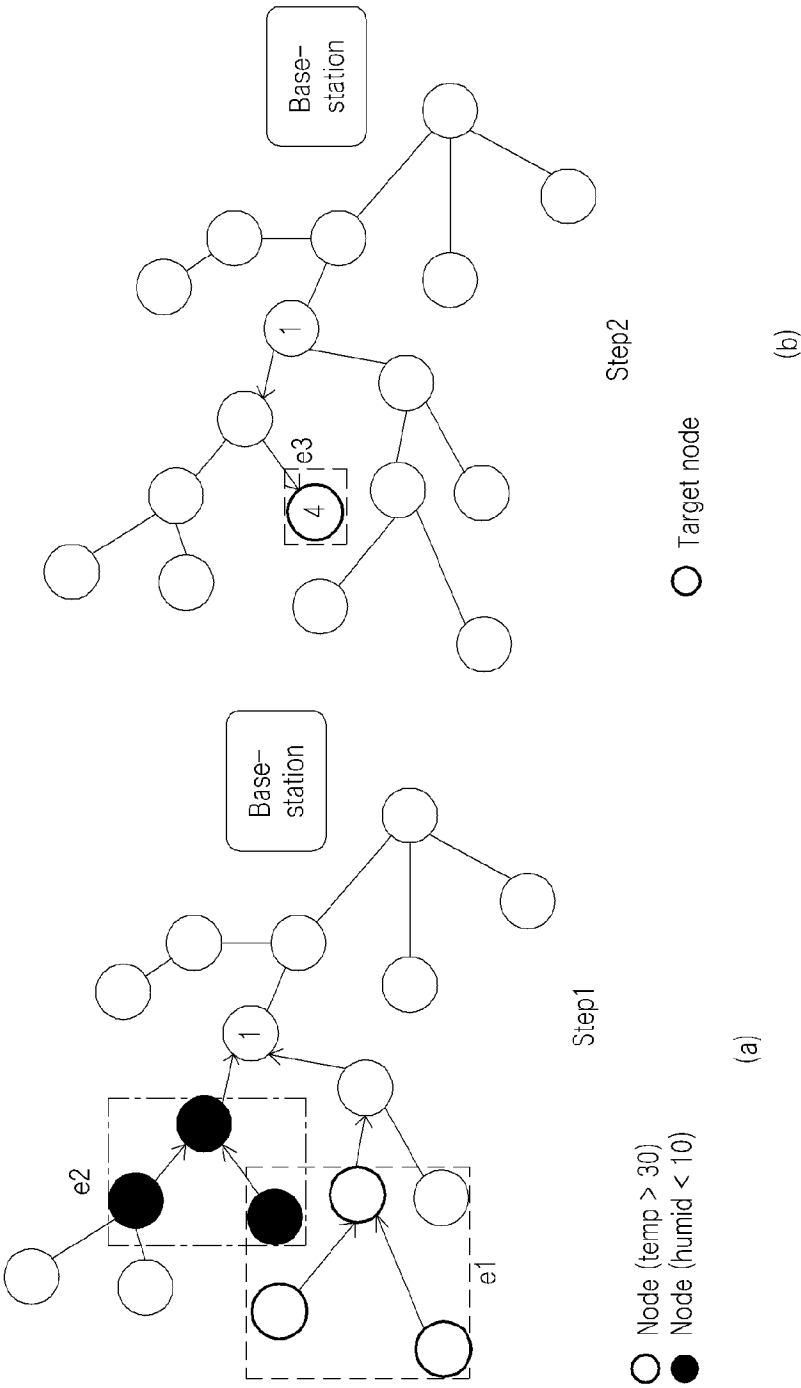
○ Node (temp > 30)

(a)

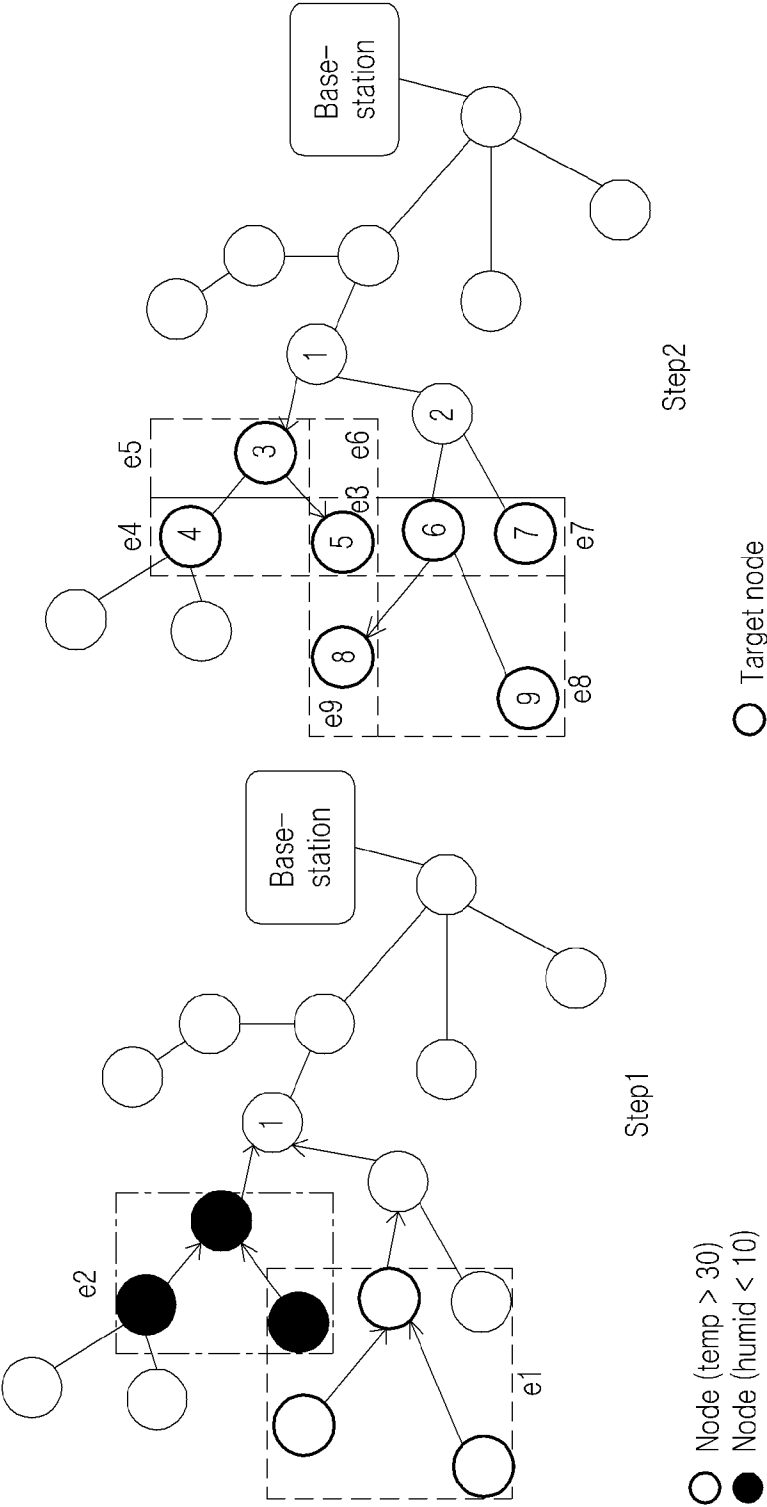
[도6]



[도7]



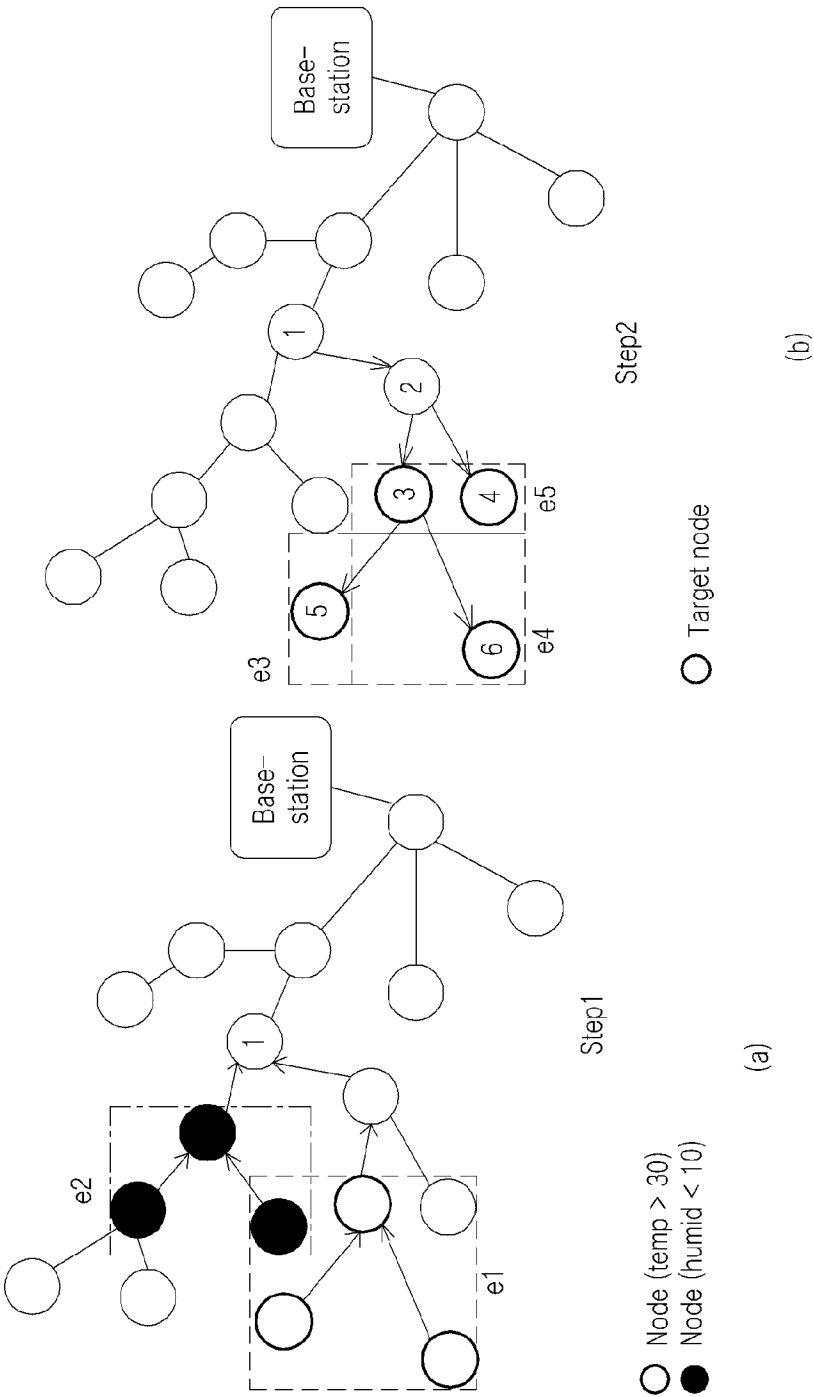
[도8]



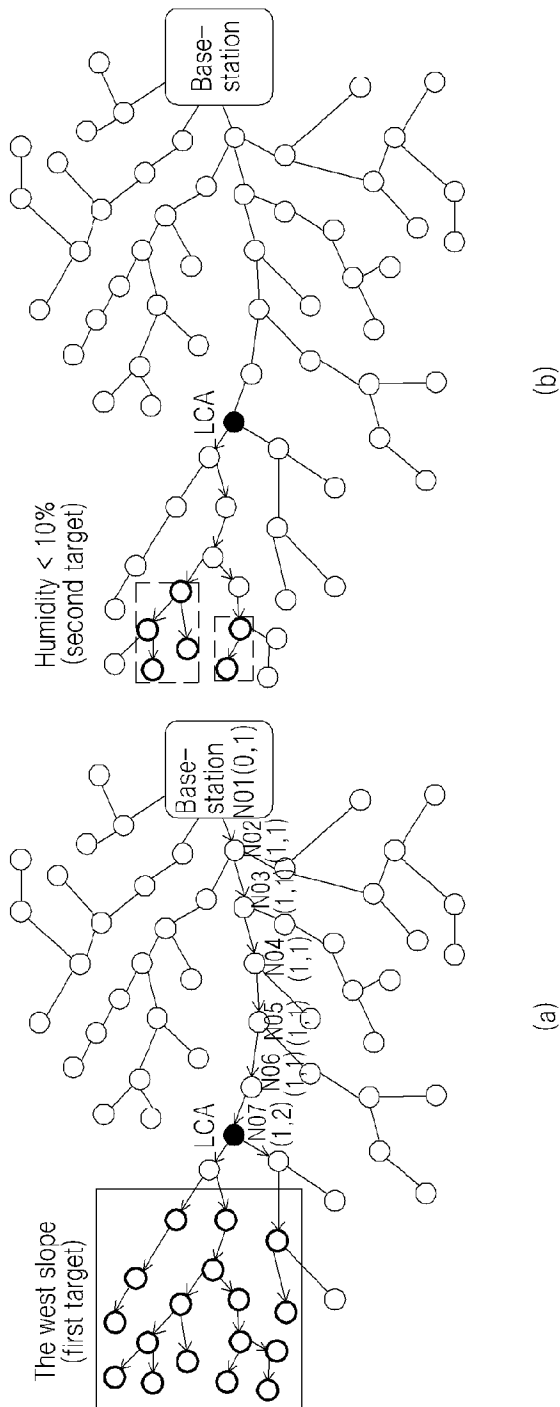
(a)

(b)

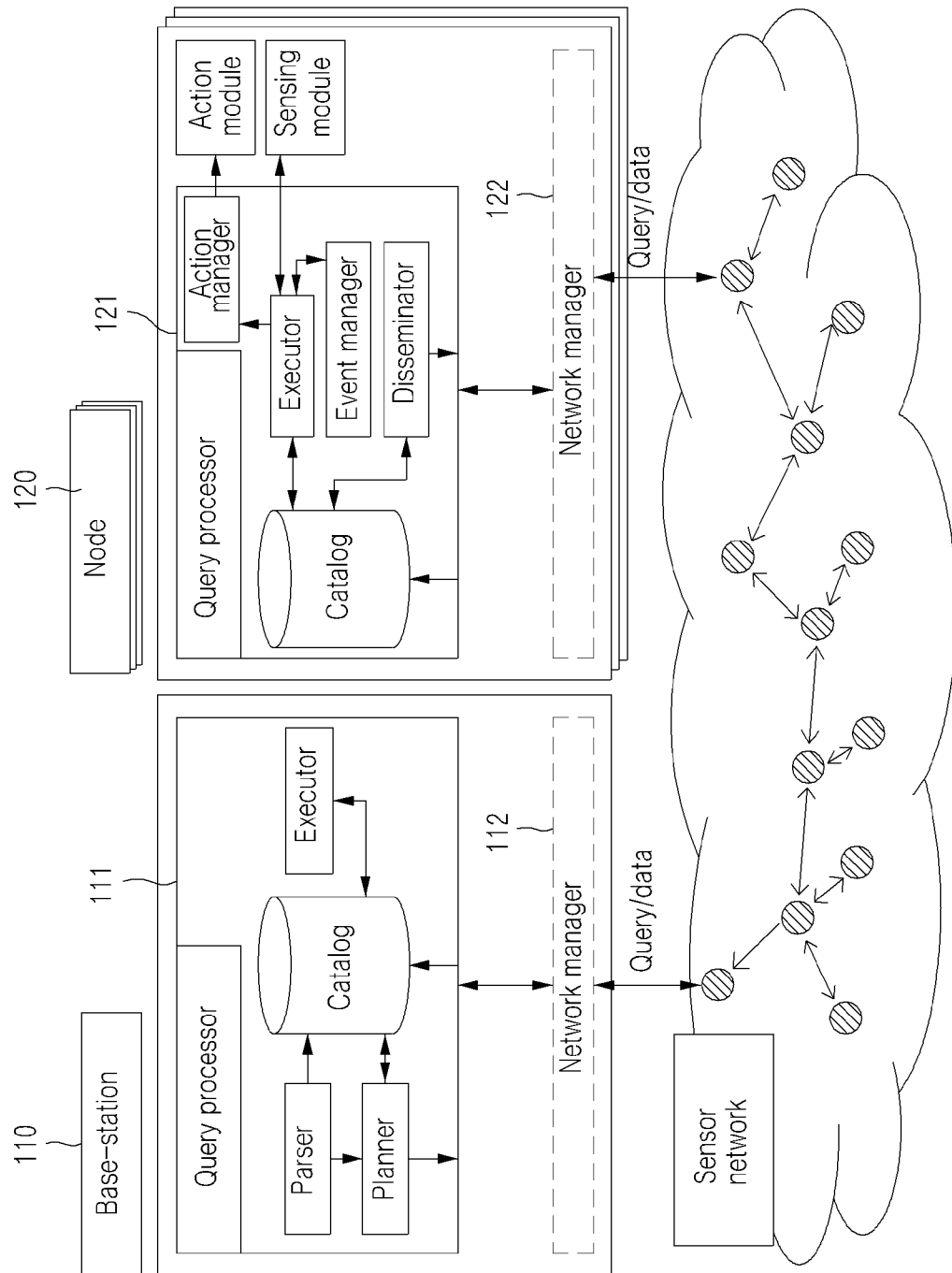
[도9]



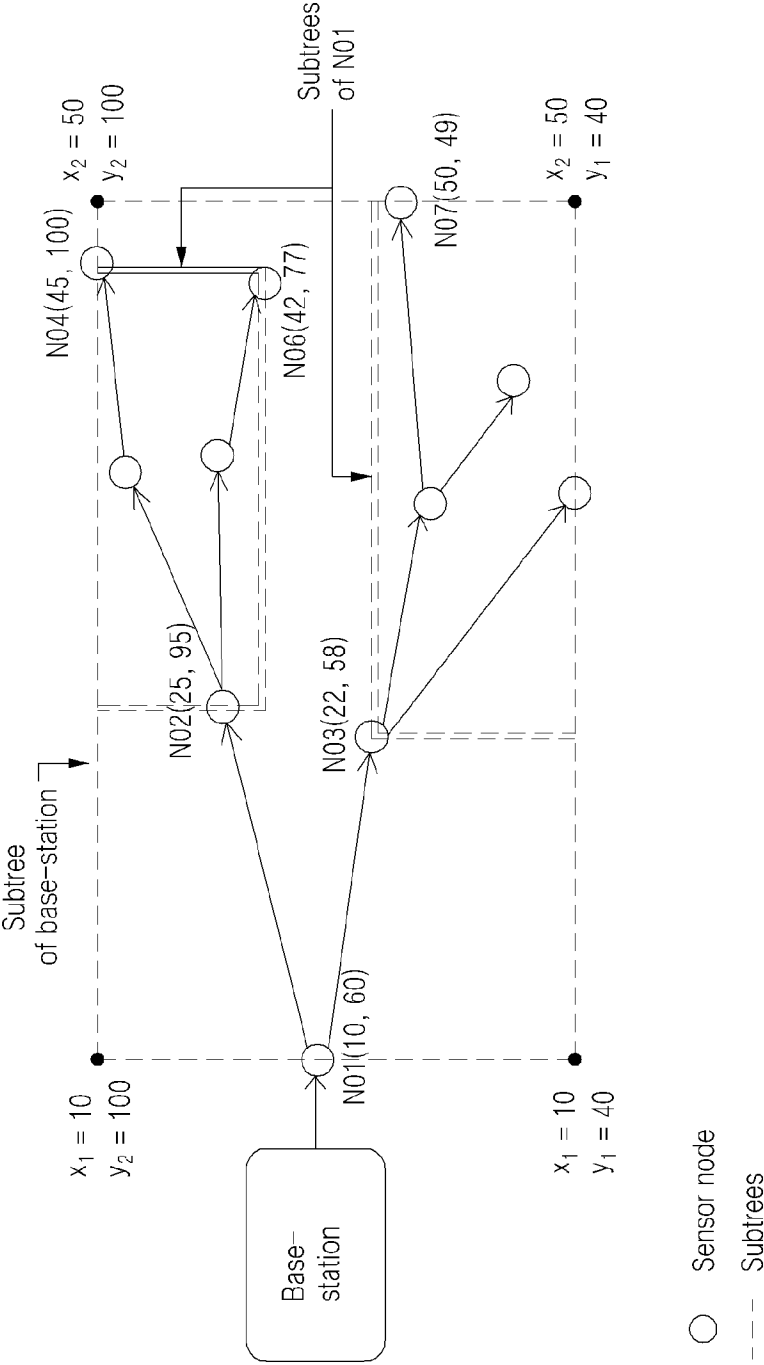
[도10]



[도11]



[도12]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2016/000852

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/24(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L 12/24; G06F 17/10; G06F 15/16; G06F 17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean Utility models and applications for Utility models: IPC as above

Japanese Utility models and applications for Utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & Keywords: wireless sensor network, query SNQL, region, spatial operator

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PARK, Min Jee et al., "Efficient Space Assignment Query Language and Processing Scheme in Sensor Database: An Explorative Stud", Journal of Korean Institute of Information Scientists and Engineers, vol. 40, no. 1, pages 36-42, February 2013	1-4,7-12,15-16
Y	See section 1, 2, 3.3, 4.1-4.2; and table 1.	5-6,13-14
Y	LIM, Chong Sok et al., "Design and Implementation of Spatial Operators and Energy-Efficient Query Processing Strategy in Wireless Sensor Network Database System", International Journal of Distributed Sensor Networks - Special Issue on Mobile Sensing and Data Management for Sensor Networks 2014, Volume 2015, Article No. 7, pp. 1-17, January 2015 See section 4; and figure 11. * The above document(thesis) is a document declaring exceptions to lack of novelty when the present PCT application is filed.	5-6,13-14
X	LIM, Jong Seok, "Spatial Operators and Energy-Efficient Query Processing Strategy in Wireless Sensor Network Database System", Doctoral Thesis, Graduate School of Korea Advanced Institute of Science and Technology, Department of Information and Communication Engineering, February 2015 See page 1, lines 17-21; page 4, lines 1-15; page 5 lines 3-17; page 64, lines 9-14; page 65, lines 1-3; page 66, lines 1-8; and figures 10-13, 39, 41. * The above document(thesis) is a document declaring exceptions to lack of novelty when the present PCT application is filed.	1-16



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

26 MAY 2016 (26.05.2016)

Date of mailing of the international search report

26 MAY 2016 (26.05.2016)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Seonsa-ro, Daejeon 302-701,
Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2016/000852

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-1480261 A (KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY) 12 January 2015 See paragraphs [0002]-[0023]; and figure 1.	1-16
A	KR 10-2012-0097302 A (INDUSTRY-ACADEMIC COOPERATION FOUNDATION, YONSEI UNIVERSITY) 03 September 2012 See paragraphs [0002]-[0018]; and figure 1.	1-16
A	US 2006-0282498 A1 (MURO, Keiro) 14 December 2006 See paragraphs [0006]-[0014]; and figure 2.	1-16

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2016/000852

Patent document cited in search report	Publication date	Patent family member	Publication date
KR 10-1480261 B1	12/01/2015	NONE	
KR 10-2012-0097302 A	03/09/2012	KR 10-1237736 B1	26/02/2013
US 2006-0282498 A1	14/12/2006	CN 1878065 A	13/12/2006
		CN 1878065 B	12/05/2010
		JP 2006-344017 A	21/12/2006
		KR 10-2006-0128597 A	14/12/2006

A. 발명이 속하는 기술분류(국제특허분류(IPC))

H04L 12/24(2006.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)

H04L 12/24; G06F 17/10; G06F 15/16; G06F 17/30

조사된 기술분야에 속하는 최소문헌 이외의 문헌

한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))

eKOMPASS(특허청 내부 검색시스템) & 키워드: 무선 센서 네트워크, 쿼리, SNQL, 영역, 공간 연산자

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
X	박민지 등, `센서 데이터베이스에서 효율적 공간지정 질의 언어 및 처리 기법`, 정보과학회논문지, 제40권 제1호, 페이지 36-42, 2013.02	1-4, 7-12, 15-16
Y	섹션 1, 2, 3.3, 4.1-4.2; 및 표 1 참조.	5-6, 13-14
Y	CHONG SOK LIM et al., `Design and Implementation of Spatial Operators and Energy-Efficient Query Processing Strategy in Wireless Sensor Network Database System`, International Journal of Distributed Sensor Networks - Special issue on Mobile Sensing and Data Management for Sensor Networks 2014, Volume 2015, Article No. 7, pp. 1-17, 2015.01	5-6, 13-14
	섹션 4; 및 도면 11 참조. ※ 위 문헌(논문)은 본 출원의 우선권에서 공지예외 주장된 문헌임	
X	임종석, `무선 센서네트워크 데이터베이스 시스템에서 공간질의를 위한 오퍼레이터 및 에너지 효율적인 질의처리 기법`, 박사학위 논문, 한국과학기술원 대학원, 정보통신공학과, 2015.02	1-16
	페이지 1, 라인 17-21; 페이지 4, 라인 1-15; 페이지 5 라인 3-17; 페이지 64, 라인 9-14; 페이지 65, 라인 1-3; 페이지 66, 라인 1-8; 및 도면 10-13, 39, 41 참조. ※ 위 문헌(논문)은 본 출원의 우선권에서 공지예외 주장된 문헌임	

☒ 추가 문헌이 C(계속)에 기재되어 있습니다.☒ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

“A” 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌

“T” 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌

“E” 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌

“X” 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.

“L” 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌

“Y” 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.

“O” 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌

“&” 동일한 대응특허문헌에 속하는 문헌

“P” 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

국제조사의 실제 완료일

2016년 05월 26일 (26.05.2016)

국제조사보고서 발송일

2016년 05월 26일 (26.05.2016)

ISA/KR의 명칭 및 우편주소

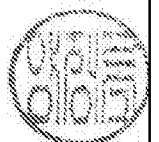
대한민국 특허청
(35208) 대전광역시 서구 청사로 189,
4동 (둔산동, 정부대전청사)

팩스 번호 +82-42-481-8578

심사관

양정록

전화번호 +82-42-481-5709



C(계속). 관련 문헌		
카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	KR 10-1480261 A (한국과학기술원) 2015.01.12 단락 [0002]-[0023]; 및 도면 1 참조.	1-16
A	KR 10-2012-0097302 A (연세대학교 산학협력단) 2012.09.03 단락 [0002]-[0018]; 및 도면 1 참조.	1-16
A	US 2006-0282498 A1 (KEIRO MURO) 2006.12.14 단락 [0006]-[0014]; 및 도면 2 참조.	1-16

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
KR 10-1480261 B1	2015/01/12	없음	
KR 10-2012-0097302 A	2012/09/03	KR 10-1237736 B1	2013/02/26
US 2006-0282498 A1	2006/12/14	CN 1878065 A	2006/12/13
		CN 1878065 B	2010/05/12
		JP 2006-344017 A	2006/12/21
		KR 10-2006-0128597 A	2006/12/14