

<https://doi.org/10.7236/IIBC.2017.17.1.139>

IIBC 2017-1-18

센서 네트워크에서 다차원 데이터를 위한 쿼리 처리 시스템

Query Processing System for Multi-Dimensional Data in Sensor Networks

김장수*, 김정준**, 김영곤***, 이창훈****

Jang-Soo Kim*, Jeong-Joon Kim**, Young-Gon Kim***, Chang-Hoon Lee****

요약 최근 GeoSensor 활용이 증가함에 따라 공간 센서 데이터와 같은 2차원 데이터를 효율적으로 처리하기 위한 질의 처리 시스템이 활발히 연구되고 있다. 하지만 기존 질의 처리 시스템들은 시공간 센서 데이터와 같은 다차원 데이터를 처리하기 위한 데이터 타입과 연산자를 지원하지 않기 때문에 이와 같은 다차원 데이터를 처리하기에 미흡하다. 따라서, 본 논문은 이러한 센서 네트워크에서 다차원 데이터를 효율적으로 처리하기 위하여 질의 처리 시스템을 개발하였다. 마지막으로 본 논문은 시나리오를 통해 본 시스템의 효용성을 검증하고, 수행시간 및 메모리 사용량 등의 성능 평가를 통해 기존 시스템들보다 성능이 우수함을 입증하였다.

Abstract As technologies related to sensor network are currently emerging and the use of GeoSensor is increasing along with the development of IoT technology, spatial query processing systems to efficiently process spatial sensor data are being actively studied. However, existing spatial query processing systems do not support a spatial-temporal data type and a spatial-temporal operator for processing spatial-temporal sensor data. Therefore, they are inadequate for processing spatial-temporal sensor data like GeoSensor. Accordingly, this paper developed a spatial-temporal query processing system, for efficient spatial-temporal query processing of spatial-temporal sensor data in a sensor network. Lastly, this paper verified the utility of System through a scenario, and proved that this system's performance is better than existing systems through performance assessment of performance time and memory usage.

Key Words : Sensor Networks, Sensor Query Processing System, Multi-Dimensional Data Type, Multi-Dimensional Operator

1. 서론

최근에 센서 네트워크에서 센서 데이터에 대한 질의 처리를 하기 위해 TinyDB, TikiriDB 등과 같은 센서 질의

처리 시스템들이 연구되었다^[1,6]. 더 나아가 GeoSensor와 같은 2차원 공간 센서 데이터에 대한 질의 처리를 위해 TinyDB를 2차원 질의 처리가 가능하도록 확장한 SE TinyDB(Spatial Extension of TinyDB)와 Spatial

*정회원, 정보시스템관리

**정회원, 한국산업기술대학교 컴퓨터공학과

***정회원, 한국산업기술대학교 컴퓨터공학과

****정회원, 건국대학교 컴퓨터공학과

접수일자 2016년 12월 2일, 수정완료 2017년 1월 12일

게재확정일자 2017년 2월 3일

Received: 2 December, 2016 / Revised: 12 January, 2017 /

Accepted: 3 February, 2017

**Corresponding Author: jkim@kpu.ac.kr

Dept. of Computer Engineering, Korea Polytechnic University, Korea

TinyDB 그리고 TikiriDB를 2차원 질의 처리가 가능하도록 확장한 SE TikiriDB(Spatial Extension of TikiriDB)와 같은 질의 처리 시스템이 연구되었다^[2,5]. GeoSensor에서는 2차원 공간 센서 데이터뿐만 아니라 다차원 데이터인 시공간 센서 데이터도 수집된다. 하지만 기존 질의 처리 시스템들은 이러한 다차원 데이터를 처리하기 위한 데이터 타입과 연산자를 지원하지 않기 때문에 다차원 센서 데이터를 처리하는 데 미흡하다.

따라서, 본 논문에서는 이러한 센서 네트워크에서 다차원 데이터에 대해 질의 처리를 효율적으로 제공하기 위한 STTinyDB(Spatio-Temporal TinyDB)를 개발하였다. 본 논문에서 개발하는 STTinyDB는 다차원 센서 데이터를 처리하기 위해 대표적인 센서 질의 처리 시스템인 TinyDB를 확장하여 개발하며, 상호운용성을 위해서 국제공간정보표준화기구 OGC(Open Geospatial Consortium)에서 제시한 “Simple Features Specification for SQL” 표준을 확장하여 구현한다^[3,4]. 본 논문의 STTinyDB에 대한 전체 시스템 구성은 인터페이스 관리자와 데이터 통신 관리자, 다차원 데이터 관리자, 다차원 연산 관리자, 다차원 데이터 스트림 관리자로 이루어져 있다.

II. 관련 연구

1. Spatial TinyDB

USN(Ubiquitous Sensor Network) 환경에서 u-GIS, u-LBS, u-Learning, u-City, u-교통, u-물류 등과 관련된 다양한 서비스를 제공하기 위해 TinyDB를 확장하여 2차원 공간 데이터를 효율적으로 처리할 수 있는 Spatial TinyDB를 개발하였다^[5]. Spatial TinyDB는 TinyDB에 OGC에서 제시한 “Simple Features Specification for SQL” 표준 명세를 따르는 2차원 공간 데이터 타입과 공간 연산자를 제공한다. 표 1은 Spatial TinyDB에서 제공하는 공간 데이터 타입의 예를 보여준다.

표 1. 공간 데이터 타입

Table 1. Spatial Data Type

Data Type	Example
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 1에서 보는 바와 같이 2차원 공간 데이터 타입으로는 point, lineString, polygon, polyhedralSurface, multiPoint, multiLineString, multiPolygon 등의 데이터 타입 7개를 지원한다. 표 2는 센서 노드에서 질의 처리를 위해 Spatial TinyDB에서 제공하는 연산자의 종류를 보여준다.

표 2. Spatial TinyDB 연산자

Table 2. Operator of Spatial TinyDB

Operator Type	Operator
Relation perator	Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects, Contains
Analysis Operator	Distances, Intersection, Difference, Union, Buffer, ConvexHull
Trajectory Operator	Enters, Insides, Leaves, Meets, Passes

표 2에서 보는 바와 같이 관계 연산자 8개, 분석 연산자 6개, 궤적 연산자 5개를 지원한다.

2. SE TikiriDB

센서 네트워크에서 2차원 공간 데이터 질의를 처리하기 위해서 TikiriDB를 확장하여 SE TikiriDB를 개발하였다^[6]. SE TikiriDB는 TikiriDB에 OGC에서 제시한 “Simple Features Specification for SQL” 표준 명세를 따르는 데이터 타입과 연산자를 일부분 제공한다. 표 3은 SE TikiriDB에서 제공하는 연산자의 종류를 보여준다.

표 3. SE TikiriDB 연산자

Table 3. Operator of SE TikiriDB

Operator	Explain
Distance	Distance(geometry A, geometry B) : Double
Disjoint	Disjoint(geometry A, geometry B) : Boolean
Contains	Contains(geometry A, geometry B) : Boolean
Within	Within(geometry A, geometry B) : Boolean

표 3에서 보는 바와 같이 Distance 연산자는 공간 객체 A와 B의 좌표 거리를 반환하고, Disjoint 연산자는 공간 객체 A와 B가 공간적으로 만나지 않았는지 여부를 반환한다. Contains 연산자는 공간 객체 A가 공간 객체 B를 공간적으로 포함하는지 여부를 반환하며, Within 연산자는 공간 객체 B가 공간 객체 A를 공간적으로 포함하는지 여부를 반환한다.

III. 시스템 설계 및 구현

본 논문에서 개발한 STTinyDB는 센서 네트워크에서 군사, 의료, 기상, 환경, 교통, 가정, 회사 등과 관련된 다양한 서비스를 제공하기 위해서 TinyDB를 확장하여 다차원 센서 데이터를 효율적으로 처리하는 질의 처리 시스템이다. 그림 1은 본 논문에서 설계한 STTinyDB의 전체 시스템 구조도를 보여준다.

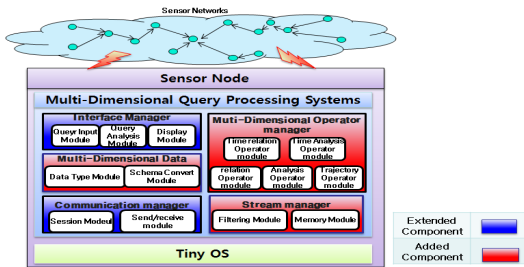


그림 1. 시스템 구조도
 Fig. 1. Architecture of System

그림 1에서 보는 바와 같이 STTinyDB는 데이터 통신 관리자, 데이터 스트림 관리자, 다차원 데이터 관리자, 인터페이스 관리자, 다차원 연산 관리자로 구성된다.

인터페이스 관리자는 사용자로부터 질의를 입력받는 질의 입력 모듈, 질의를 토큰/구문 분석하는 질의 분석 모듈, 각 센서 노드에서 처리된 질의의 최종 결과를 보여주는 디스플레이 모듈로 구성된다.

데이터 통신 관리자는 사용자가 입력한 영역에 질의 하기 위해서 해당 영역의 센서 노드들 간 세션을 관리하는 세션 관리 모듈, 해당 영역에 있는 센서 노드들 간 통신을 관리하는 데이터 전송/수신 모듈로 구성된다.

데이터 스트림 관리자는 질의를 처리할 때 끊임없이 삽입되는 다차원 센서 데이터로 인한 입력 부하를 줄이기 위해서 다양한 여과 조건을 적용하여 필터링하는 필터링 처리 모듈, 여러 개의 질의들 간 메모리 효율을 높이기 위해서 메모리를 공유할 수 있도록 관리하는 메모리 관리 모듈로 구성된다.

다차원 데이터 관리자는 OGC “Simple Features Specification for SQL” 표준의 2차원 공간 데이터 타입을 기반으로 하는 다차원 데이터 타입 모듈, TinyDB에서 제공하는 속성 정보를 다차원 스키마 매핑 룰(Mapping Rule)에 따라 다차원 속성으로 빠르게 변환하는 다차원 스키마 변환 모듈로 구성된다.

다차원 연산 관리자는 시간 연산 처리를 하는 시간 관계/분석 연산자 처리 모듈, 다차원 시공간 연산 처리를 위한 시공간 관계/분석 연산자 처리 모듈, 이동하는 센서 노드의 이동 궤적을 처리하는 시공간 궤적 연산자 처리 모듈로 구성된다.

생태계 모니터링 서비스에 대한 가상 시나리오는 철새에 의한 조류독감 확산 방지를 위해서 철새 100마리에 각각 센서를 부착하여 철새의 고유 번호로 철새의 체온과 시공간 정보를 수집하는 것이다. 표 4은 가상 시나리오에서 사용하는 질의를 보여준다.

표 4. 질의예
 Table 4. Example of Query

Query Type	Example
Query 1	SELECT nodeid, temp, stLoc FROM sensors WHERE stEnters(stMultiPolygon(stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 0 600, 0 900, 400 900 0, 400 600, 0 600)), stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 50 0, 500 300, 900 300, 900 0, 500 0)), stLoc) AND temp > 40 SAMPLE PERIOD 1024
Query 2	SELECT nodeid, temp, stLoc FROM sensors WHERE stLeaves(stMultiPolygon(stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 0 600, 0 900, 400 900 0, 400 600, 0 600)), stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 50 0, 500 300, 900 300, 900 0, 500 0)), stLoc) AND temp > 40 SAMPLE PERIOD 1024

표 4에서 보는 바와 같이 질의 1은 하루 동안 낙동강 서식지 또는 압록강 서식지 안으로 들어가는 철새 중 체온이 40°C를 초과한 철새의 체온과 다차원 시공간 정보를 반환한다. 질의 2는 하루 동안 낙동강 서식지 또는 압록강 서식지 밖으로 벗어나는 철새 중 체온이 40°C를 초과한 철새의 체온과 다차원 시공간 정보를 반환한다.

그림 2는 질의 1에 대한 결과 화면을 보여준다.

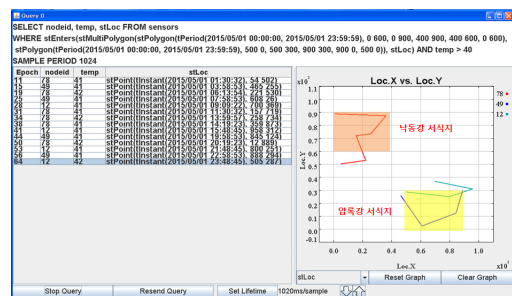


그림 2. 질의 예 1
 Fig. 2. Screen of Query 1

그림 2는 질의 1에 만족하는 철새의 체온과 시공간 정보를 보여준다. 철새의 고유 번호 중 78, 49, 12는 낙동강 서식지(stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 0 600, 0 900, 400 900, 400 600, 0 600)) 또는 압록강 서식지(stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 50 0, 500 300, 900 300, 900 0, 500 0)) 안으로 들어가는 철새들 가운데 체온이 40°C를 초과한 것을 볼 수 있다.

그림 3은 질의 2에 대한 결과 화면을 보여준다.

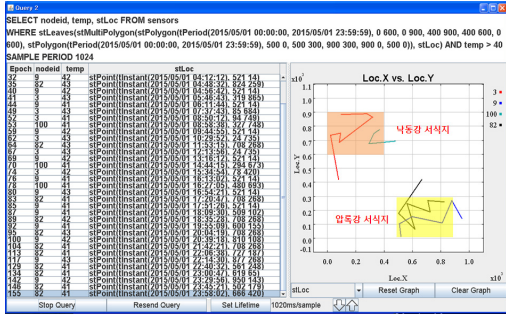


그림 3. 질의 예 2
Fig. 3. Screen of Query 2

그림 3은 질의 2에 만족하는 철새의 체온과 시공간 정보를 보여준다. 철새의 고유 번호 중 3, 9, 100, 82는 낙동강 서식지(stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 0 600, 0 900, 400 900, 400 600, 0 600)) 또는 압록강 서식지(stPolygon(tPeriod(2015/05/01 00:00:00, 2015/05/01 23:59:59), 50 0, 500 300, 900 300, 900 0, 500 0)) 밖으로 벗어나는 철새들 가운데 체온이 40°C를 초과한 것을 볼 수 있다.

본 논문에서 개발한 STTinyDB를 생태계 모니터링 시나리오에 적용하여 검증함으로써 본 시스템이 센서 네트워크에서 다차원 센서 데이터 질의 처리가 필요한 응용 분야에 유용하게 사용될 수 있음을 보였다.

IV. 실험 및 결과

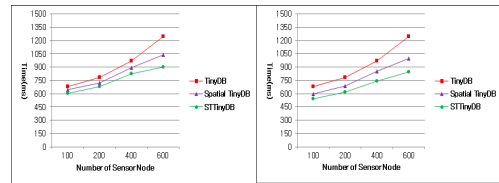
본 논문에서 개발한 STTinyDB에 대한 성능 평가를 위해 시스템 구현에서 기술한 구현 환경을 바탕으로 센서 노드 개수, 질의 개수, 데이터 생성 주기, 필터링하는 조건 등을 파라미터로 설정하여 성능을 비교하였다. 표 5는 성능 평가를 위해 설정한 파라미터를 보여준다.

표 5. 실험평가 파라미터

Table 5. Parameter for Performance evaluation

Parameter	Value
Number of Sensor Node	100, 200, 400, 600
Number of Query	20, 40, 60, 80
Period of Data Generation	128ms
Filtering Condition	Period of Filtering 1024ms
Query	TinyDB SELECT nodeid, time, x, y FROM sensors WHERE time >= "2015/04/01 09:00:00" AND time <= "2015/04/01 12:00:00" AND x >= 0 AND x <= 400 AND y >= 0 AND y <= 400
	Spatial TinyDB SELECT nodeid, time, loc FROM sensors WHERE time >= "2015/04/01 09:00:00" AND time <= "2015/04/01 12:00:00" AND contains(polygon(0 0, 0 400, 400 400, 400 0, 0 0), loc)
	STTinyDB SELECT nodeid, stLoc FROM sensors WHERE stContains(stPolygon(tPeriod(2015/04/01 09:00:00, 2015/04/01 12:00:00), 0 0, 0 400, 400 400, 400 0, 0 0), stLoc)

수행 시간 평가는 데이터 필터링을 사용했을 경우와 필터링을 사용하지 않았을 경우로 구분해서 하나의 다차원 시공간 질의에 대해 각 센서 노드에서 질의를 처리하고, 반환된 평균 시간을 비교하였다. 센서 노드의 수는 100개부터 600개까지 늘려가면서 센서 노드의 증가로 발생하는 성능 향상을 살펴보았다. 그림 4는 센서 노드 개수에 따른 수행 시간을 측정된 결과를 보여준다.



(a) Used Filtering (b) Non Used Filtering

그림 4. 센서 노드 개수에 따른 수행 시간

Fig. 4. Execution Time by Sensor Node Number

그림 4 (a)에서 보는 바와 같이 필터링 미적용 시 본 논문에서 제안한 STTinyDB가 TinyDB에 비해 평균 14%, Spatial TinyDB에 비해 평균 7% 성능이 향상된 것

을 보여준다. 그리고 그림 4 (b)에서 보는 바와 같이 필터링 적용 시 STTinyDB가 TinyDB에 비해 평균 28%, Spatial TinyDB에 비해 평균 14% 성능이 향상된 것을 보여준다. 이러한 성능 평가 결과가 나타나는 이유는 본 논문에서 제안한 STTinyDB의 다차원 데이터 타입과 연산자, 필터링 처리 기능으로 다차원 시공간 질의를 보다 빠르게 처리하기 때문이다. 다시 말해서 TinyDB는 where 절에서 time, x, y 속성을 나눠서 처리하고, Spatial TinyDB는 where 절에서 time 속성을 따로 처리하는 반면 STTinyDB는 연산자 하나로 한 번에 처리하기 때문에 비교 과정이 적게 들어 다른 두 시스템보다 빠르다.

메모리 사용량 평가는 데이터 필터링을 사용했을 경우와 필터링을 사용하지 않았을 경우로 구분해서 여러 개의 다차원 시공간 질의를 하나의 센서 노드에서 처리할 경우 여러 개의 질의들 간 메모리를 얼마만큼 공유하여 메모리 사용량을 줄이는 지를 비교하였다. 다차원 시공간 질의 개수는 20개부터 80개까지 늘려가면서 다차원 시공간 질의의 증가로 발생하는 메모리 사용 효율을 살펴봐왔다. 그림 5는 질의 개수에 따른 메모리 사용량을 측정된 결과를 보여준다.

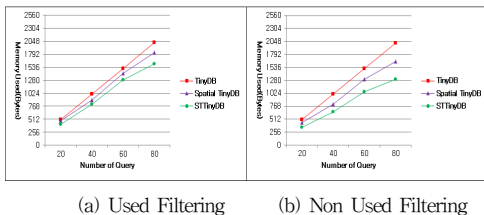


그림 5. 질의 수에 따른 메모리 사용량
 Fig. 5. Memory Use Rate by Query Number

그림 5 (a)에서 보는 바와 같이 필터링 미적용 시 본 논문에서 제안한 STTinyDB가 TinyDB에 비해 평균 11%, Spatial TinyDB에 비해 평균 5% 메모리 사용이 줄어든 것을 보여준다. 그리고 그림 5 (b)에서 보는 바와 같이 필터링 적용 시 STTinyDB가 TinyDB에 비해 평균 24%, Spatial TinyDB에 비해 평균 10% 메모리 사용이 줄어든 것을 보여준다. 이러한 성능 평가 결과가 나타나는 이유는 본 논문에서 제안한 STTinyDB의 메모리 공유 기능으로 인해 여러 개의 질의가 메모리를 공유하여 사용하기 때문이다.

V. 결론

최근 IoT 기술의 발전과 더불어 다양한 센서 데이터를 수집하는 기술과 무선 통신 기술이 급격히 발전함에 따라 센서 네트워크 관련 기술에 대한 관심과 연구가 증대되고 있다.

센서 네트워크에서 2차원 공간 센서 데이터에 대한 효율적인 질의 처리를 위한 여러 가지의 공간 질의 처리 시스템이 연구되었다. 그러나, 기존 공간 질의 처리 시스템들은 다차원 센서 데이터 처리를 위한 데이터 타입과 연산자를 지원하지 않기 때문에 다차원 센서 데이터를 처리하는 데 미흡했다.

따라서, 이러한 문제를 해결하기 위해서 본 논문에서는 OGC의 표준 명세 중 “Simple Features Specification for SQL”을 확장하여 다차원 시공간 질의 처리 시스템을 개발하였다. 본 시스템은 다차원 데이터 타입과 시공간 관계/분석 연산자를 제공하고, 이동 센서의 궤적을 처리하기 위해서 궤적 연산자도 제공한다. 또한, 본 시스템에서 사용되는 다양한 다차원 시공간 질의들 간 필요한 데이터를 공유하여 데이터 스트림으로 인해 발생하는 시스템 부하를 줄이도록 메모리 공유 기능을 제공하였으며, 처리하지 못할 많은 입력 데이터 스트림이 들어왔을 경우 최소한의 정확도 손실로 입력 데이터 스트림 양을 최대한 줄여 과부하가 발생하지 않도록 필터링하는 기능을 제공하였다.

마지막으로, 본 논문에서 개발한 STTinyDB를 생체계 모니터링 시나리오에 적용하여 검증함으로써 본 시스템이 센서 네트워크에서 다차원 시공간 질의 처리가 필요한 많은 응용 분야에 유용하게 사용될 수 있음을 보였다. 그리고 STTinyDB 및 TinyDB, Spatial TinyDB와 비교 성능 평가를 통해 STTinyDB가 다차원 시공간 연산자와 필터링 처리 기능으로 인해 다른 두 시스템보다 질의를 처리하는 데 더 빠른 성능을 보였으며, 메모리 공유 기능으로 인해 다른 두 시스템보다 메모리를 효율적으로 사용함을 알 수 있었다. 전체적으로 본 논문에서 제안한 STTinyDB는 다른 두 시스템보다 성능이 우수함을 확인할 수 있었다.

본 논문에서는 시간 변화에 따른 다차원 센서 데이터의 검색을 위한 다차원 인덱스 기법을 고려하지 못했다. 따라서, 향후 연구로 다차원 센서 데이터의 빠른 질의 처리를 위하여 다차원 인덱스의 적용 방안에 대한 연구가 필요하다.

References

- [1] Ale, R., "Working with Spatio-Temporal Data Type," Proc. of the Int. Conf. on Society for Photogrammetry and Remote Sensing, 2012, pp.5-10.
- [2] Kim, D. O., Liu, L., Shin, I. S., Kim, J. J., and Han, K. J., "Spatial TinyDB: A Spatial Sensor Database System for the USN Environment," Journal of Distributed Sensor Networks, 2013.
DOI: <https://doi.org/10.1155/2013/512368>
- [3] Open Geospatial Consortium, "OpenGIS Implementation Specification for Geographic Information-Simple Feature Access-Part 1:Common Architecture," Version 1.2.1, 2011.
DOI: <https://doi.org/10.3403/03212261>
- [4] Open Geospatial Consortium, "OpenGIS Implementation Specification for Geographic Information-Simple Feature Access-Part 2:SQL Option," Version 1.2.1, 2011.
DOI: <https://doi.org/10.3403/bseniso19125>
- [5] Park, J., Kim, K., Ahn, S., and Hong, B., "Continuous Query Processing on Data Stream: Sensor, Location and Identification," Proc. of Seventh Int. Conf. on Information Technology, 2010, pp.518-522.
DOI: <https://doi.org/10.1109/itng.2010.204>
- [6] Hyeun Kim, Do-Hyeun Kim, Hee-Do Park, "A Tracking Service of Animal Situation using RFID, GPS, and Sensor," Journal of IIBC, Vol. 9. No. 5, pp.79-84.

저자 소개

김 장 수(정회원)



• Jang Soo Kim received his BS in Business Administration at Kyungpook University in 1983 and MS in Graduate School of Public Policy at Sogang University in 1997, respectively. He is currently doctoral Candidate at the department of Computer Science at Konkuk University.

He work currently at a Information System Supervision. His research interests include Database Systems, Software Engineering, Information System Supervision etc.

김 정 준(정회원)



• Jeong Joon Kim received his BS and MS in Computer Science at Konkuk University in 2003 and 2005, respectively. In 2010, he received his PhD in at Konkuk University. He is currently a professor at the department of Computer Science at Korea Polytechnic University.

His research interests include Database Systems, BigData, Semantic Web, Geographic Information Systems (GIS) and Ubiquitous Sensor Network (USN), etc.

김 영 곤(정회원)



• Young Gon Kim received his BS in Electronic Engineering at Kyungpook University in 1983 and MS in Electronic Engineering at Yonsei University in 1985, respectively. In 2000, he received his PhD in at KAIST. He is currently a

professor at the department of Computer Science at Korea Polytechnic University. His research interests include Software Engineering, nformation communication system, object-oriented analysis and design, etc.

이 창 훈(정회원)



• Chang Hoon Lee received his BS in Department of Mathematics at Yonsei University in 1977 and MS in Computer Science at KAIST in 1980, respectively. In 1993, he received his PhD in at KAIST. He is currently a professor at the department of Computer Science at

Konkuk University. His research interests include Artificial Intelligence, Operating System, Information Security, etc.