

# 실시간 공간 상황 분석을 위한 공간 데이터 스트림 처리 시스템

## A Spatial Data Stream Processing System for Spatial Context Analysis in Real-time

권 오 제\*      김 재 훈\*\*      이 기 준\*\*\*  
Kwon O Je      Jae Hun Kim      Ki Joune Li

**요약** 센서로부터 획득되는 데이터 스트림들 중에 특히 이동 객체에 대한 공간 정보를 담은 데이터 스트림은 상황 인지의 여러 응용 분야에 매우 유용하다. 하지만, 실시간으로 공간 스트림을 처리하는 것과 공간 상황 인지를 위한 복잡한 연산 처리 사이에는 중요한 기능적인 격차가 존재하는데, 이는 공간 스트림 처리와 상황 인지를 통합하는데 매우 큰 어려움을 준다. 본 논문에서는, 공간 스트림 처리와 공간 상황 인지 사이에 존재하는 기능적인 격차를 해결하기 위한 공간 데이터 스트림 처리 시스템인 SCONSTREAM(Spatial CONtext STREAM Management)를 제안한다. 본 논문에서 제안하는 시스템은 불필요한 공간 데이터 스트림을 전처리하고 상황 인지에 용이한 형태로 가공한다. 실험을 통해, 본 시스템이 공간 스트림 처리와 공간 상황 인지 사이에 존재하는 기능적인 격차를 해소할 수 있음을 보인다.

**키워드** : 데이터 스트림, 공간 상황 인지, 실시간 처리

**Abstract** Spatial data streams from sensors are useful in context-awareness for many types of applications. However, an important gap is found between spatial data stream management in real-time and complex computation for spatial context-awareness, and this brings about serious difficulty to integrate spatial data stream processing and context-awareness. In this paper, we present a system called SCONSTREAM(Spatial CONtext STREAM Management) that we have developed to resolve the gap between spatial data stream and context-awareness. The key approach of our system is to filter off unnecessary spatial data streams and convert them to the spatial context streams, which are smaller and more suitable to be processed by the context-awareness module than raw data from sensors. By experimentation, We show that SCONSTREAM resolves the functional gap between spatial stream processing and spatial context-awareness module.

**Keywords** : data stream, spatial context-awareness, real-time processing

### 1. 서론

최근 센서 네트워크를 이용한 활발한 연구들이 진행됨에 따라, 센서로부터 다양한 종류의 데이터들을 획득하는 것이 가능해졌다. 이러한 데이터들로부터 유용한 상황들을 추출하고 분석하는 것은 유비쿼터스 컴퓨팅 환경에서 매우 근본적인 요구사항이다. 상황 정보는 사용자 기반 서비스를 제공하기 위

해 매우 신중하게 분석되어야 한다. 예를 들어, 40도 이상의 온도 정보가 센서로부터 획득되었다고 가정하자. 온도 정보로부터 분석 가능한 상황 정보는 그것이 획득된 위치가 로비이나 사우나이나에 따라 완전 다르게 분석될 수 있다.

센서로부터 획득한 공간 데이터 스트림을 처리하기 위해서는 [그림 1]의 공간 데이터 스트림 처리 모듈(Spatial Data Stream Processing Module)과

† 본 연구는 두뇌한국21사업과 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신사업과제의 연구비지원(07국토정보C04)에 의해 수행되었습니다.

\* 부산대학교 컴퓨터공학과 박사과정, kwonoje@pusan.ac.kr

\*\* 삼성전자 연구원, jhkim@pusan.ac.kr

\*\*\* 부산대학교 정보컴퓨터공학부 교수, lik@pnu.edu(교신저자)

공간 상황 인지 모듈 (Spatial Context-awareness Module)이 필요하다. 하지만 두 모듈 사이에는 몇 가지 성능적인 차이점이 존재한다.

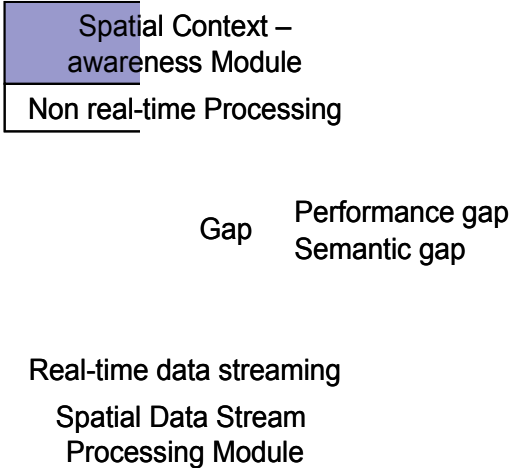


그림 1. 공간 상황 인지를 위해 필요한 두 가지 중요한 요구사항

공간 데이터 스트림 처리 모듈은 연속적으로 전달되는 대량의 데이터들을 실시간으로 처리해야 한다. 하지만 공간 상황 인지 모듈은 데이터를 분석하고 추론하는데 많은 시간이 걸린다.

공간 데이터 스트림 처리 모듈은 센서로부터 획득되는 원시 데이터를 그대로 전달해주는 것이 주된 역할인 반면, 공간 상황 인지 모듈은 의미 있는 상황 정보를 처리해야 한다.

앞서 서술한 차이점들로부터 우리는 데이터 스트림 처리와 상황 인지 사이에 존재하는 중요한 격차를 발견할 수 있다. 따라서 이러한 격차를 해결하기 위한 모듈이 필요하다.

본 논문에서는, 앞서 살펴보았던 데이터 스트림 처리와 상황 인지 사이에 존재하는 격차를 해결하기 위해 공간 데이터 처리 시스템(Spatial CONTEXT STREAM Management, SCONSTREAM) 구조를 제안한다. SCONSTREAM은 센서로부터 획득한 공간 데이터 스트림을 다음의 두 가지 이유로 공간 상황 스트림으로 변환한다.

첫 번째, 상황 인지를 위한 의미 있는 상황 정보는 센서로부터 획득되는 원시 스트림에 비해 크기가 작다. 따라서 공간 데이터 스트림을 공간 상황 스트림으로 변환하는 과정에서 스트림의 크기를 줄

임으로서 상황 인지 모듈의 처리 부하를 줄일 수 있다.

두 번째, 공간 상황 스트림이 원시 데이터 스트림에 비해 상황 인지에 보다 적합한 형태이다. SCONSTREAM은 실시간 공간 분석을 지원하도록 여러 공간 연산자들을 제공한다.

본 논문은 다음과 같이 구성된다. 2장에서는 여러 관련 연구들을 검토하고, 본 논문의 연구 동기를 서술한다. 3장에서는 공간 상황 분석을 위해 고려해야 할 공간 스트림 분석의 요구사항들에 대해 서술한다. 4장에서는 본 논문에서 제안하는 SCONSTREAM의 시스템 구조에 대해 서술한다. 그리고 여러 실현 가능한 시나리오를 바탕으로 SCONSTREAM의 성능 분석 결과를 5장에 서술한다. 마지막으로 본 논문의 결론 및 향후 연구 과제를 6장에 서술한다.

## 2. 관련 연구와 연구 동기

센서로부터 획득되는 데이터 스트림의 크기는 무한하기 때문에, 전체 데이터 스트림을 동시에 처리할 수 없다. 따라서 무한한 크기의 데이터 스트림을 유한한 크기로 나누어 연속적으로 처리하기 위한 다양한 연구들이 진행되었다. 대표적인 연구들로는 STREAM[1], TelegraphCQ[2], NiagaraCQ[3], SensorEdge[4], 그리고 Aurora[5] 가 있다.

STREAM은 기존의 SQL을 연속 질의 처리에 적합한 CQL(Continuous Query Language)로 확장하였다. TelegraphCQ는 PostgreSQL을 기반으로 대량의 센서 스트림을 처리하기 위해 개발되었다. TelegraphCQ는 스트림 처리 과정에서 발생하는 자원 소모와 성능 저하를 개선하기 위해 부하 균형 방법(load balancing method)를 제안하였다. NiagaraCQ는 인터넷 상에서 획득할 수 있는 다량의 XML 데이터 스트림을 처리하기 위해 유사한 질의들을 그룹화 하여 동시에 처리하여 질의 수를 최소화하는 방법을 제안하였다. Aurora 역시 질의 처리 성능 저하가 나타나지 않는 범위 내에서 데이터를 줄이는 부하 산개(load shedding) 방법을 제안하였다. Oracle은 다양한 응용 시스템들에게 웹과 스트림을 통해 센서 데이터를 제공하기 위한 Sensor Edge Server를 개발하였다. Sensor Edge Server는 기존의 DBMS를 사용하여 질의 결과를 제공하는

기능도 제공한다.

앞서 서술한 연구들은 단지 센서 스트림을 제공하는 기능만을 제공하고 있는 반면에, 다음의 여러 연구들은 OLAP를 기반으로 센서 데이터를 분석하기 위한 방법들을 제안하였다. MAIDS[6]는 네트워크의 통신량과 같이 전달 속도가 매우 빠른 데이터 스트림을 마이닝 하기 위한 방법을 제안하였다. Stream Cube[7]는 대량의 데이터 스트림에 대해 고단위의 동적 분석을 수행하기 위한 구조를 제안하였다. Stream Cube는 시간의 우선순위를 기반으로 데이터 스트림을 추상화하는 방법을 제안하였다.

표 1은 기존의 연구들의 성능을 비교하여 정리한 것이다. 본 논문의 목적은 데이터 스트림 처리에 상황 분석을 통합하는 것이다. 하지만 <표 1>의 기존 연구들은 Stream Cube를 제외하고 상황 분석에 대한 고려가 없다. 뿐만 아니라, Stream Cube는 공간적인 상황에 대한 분석을 고려하고 있지 않다.

### 3. 공간 상황 스트림 처리 시스템의 기본 개념

센서로부터 획득되는 공간 데이터 스트림은 위치

정보를 포함하는데, 다음과 같이 정의된다.

정의 1. 공간 데이터 스트림(Spatial Data Stream)

센서 a로부터 획득되는 공간 데이터 스트림 DS(a)은 다음과 같이 정의된다.

$$DS(a) = \{(p(a, t), t) | t \in [0, \infty)\}$$

p(a, t)는 시간 t에서 센서 a의 위치를 나타내고, 시간은 센서 데이터의 발생 시간(valid time)을 나타낸다.

센서의 위치정보는 일반적으로 x(a, t), y(a, t), z(a, t)로 표현된다. 하지만, (room\_id, t)와 같이 상징적인(symbolic) 형태로도 표현될 수 있다. 공간 데이터 스트림 처리 모듈은 오직 위치 정보를 받는 반면, 이 데이터로부터 보다 의미 있는 상황 정보를 추출해야한다. 본 논문에서는 이러한 보다 의미 있는 상황 스트림을 공간 상황 스트림이라 정의한다.

정의 2. 공간 상황 스트림(Spatial Context Stream)

센서 a로부터 획득되는 공간 스트림 데이터로부터 추출되는 공간 상황 스트림 CS(a)는 다음과 같이 정의된다.

$$CS(a) = \{(P, M, x, t) | t \in [0, \infty)\}$$

P는 공간 조건을 나타내고, M은 공간 조건 P를

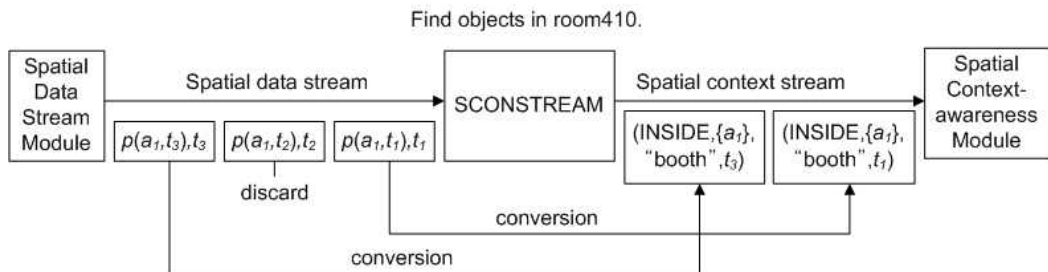


그림 2. SCONSTREAM에서 공간 데이터 스트림을 공간 상황 스트림으로 변환하는 예

표 1. 기존 연구들의 성능 비교

Systems / Function	STREAM	TelegraphCQ	NiagaraCQ	Aurora	Sensor Edge	MAIDS	Stream Cube
Sensor network	X	O	X	O	O	X	X
Real-time processing	O	O	O	O	O	O	Δ
Context-awareness	X	X	X	X	X	X	O
Spatiotemporal analysis	X	X	X	X	X	X	X

만족하는 공간 객체의 집합을 나타낸다.  $x$ 는 참조 위치를 나타내는데, 이는 공간 조건을 처리하기 위해 필요하다.

그림 2와 같이 SCONSTREAM은 공간 데이터 스트림을 연속적으로 공간 상황 스트림으로 변환한다.

예를 들어, 객체  $a$ 의 시간  $t$ 에서의 위치 정보  $p(a, t)$ 에 대한 스트림을 공간 데이터 스트림이라 할 때, “부스 안에 존재하는 모든 객체를 계속해서 찾아라.”라는 연속 질의의 결과가 공간 상황 스트림이 된다.

연속 질의를 만족하지 않는 스트림 데이터, 예를 들어 그림 2에서  $p(a1, t2)$ , 는 삭제된다. 반면에, 연속 질의를 만족하는 스트림 데이터들은 그림 3의 과정을 통해 공간 상황 스트림으로 변환된다.

SCONSTREAM에서 공간 데이터 스트림을 공간 상황 스트림으로 변환하는 과정은 다음의 두 단계로 구성된다.

연속 질의 처리 단계 - 그림 3에서 연속 질의를 처리하는 동안 질의를 만족하는 객체와 질의에 해당하는 시공간 연산자가 각각  $M, P$ 로 변환된다. 이

동 객체의 원시 위치 정보는 <표 2>에 서술되어 있는 시공간 연산자들에 의해 표현되는 의미 있는 위치 명으로 변환된다. 그림 2의 예에서, 이동 객체들은 부스 안에 있고, inside라는 키워드가 공간 조건자  $P$ 로 설정된다.

센서 카탈로그 결합(binding) 단계 - 센서 카탈로그 결합 단계는 공간 상황 스트림의 참조 위치  $x$ 를 찾는 단계이다. 참조 위치는 센서 카탈로그와의 결합을 통해 유도되어 지는데 센서 카탈로그는 각 센서와 센서가 배치된 위치 값 정보를 포함하고 있다.

연속 질의를 수행하기 위해서 <표 2>와 같은 여러 시공간 연산자들이 필요하다. 본 논문에서는 방과 작은 공간들로 구성된 실내 공간에서 발생할 수 있는 상황들에 초점을 두고 실내 공간의 공간 상황에 대한 시공간 연산자들을 정의하였다.

#### 4. SCONSTREAM의 구현

본 논문에서는 앞의 절에서 설명한 기능을 만족하기 위하여 SCONSTREAM을 개발하였다. 센서로부터 획득되는 스트림을 실시간으로 처리하기 위해

표 2. SCONSTREAM에서 제공하는 시공간 연산자들

연산자	설 명
$Inside(x,m)$	객체 $m$ 이 공간 $x$ 안에 있다.
$Outside(x,m)$	객체 $m$ 이 공간 $x$ 밖에 있다.
$WithinDist(x,m,d)$	객체 $m$ 이 객체 $x$ 와 거리 $d$ 내에 존재한다.
$Entering(x,m)$	객체 $m$ 이 공간 $x$ 안으로 들어온다.
$Leaving(x,m)$	객체 $m$ 이 공간 $x$ 밖으로 나간다.
$Staying(x,m,\delta)$	객체 $m$ 이 공간 $x$ 에 $\delta$ 시간 이상 존재한다.

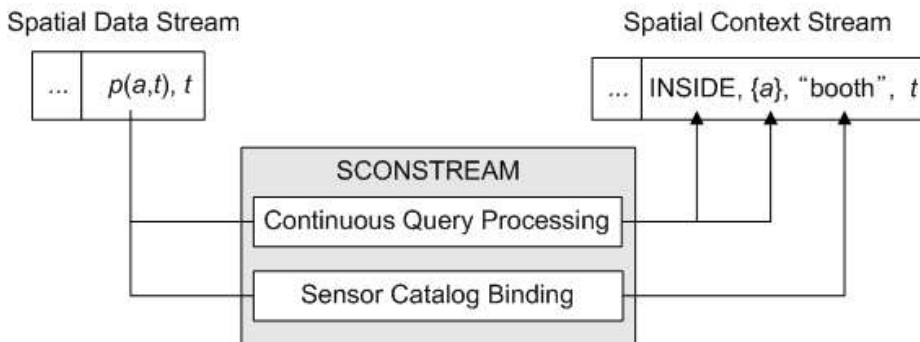


그림 3. SCONSTREAM에서 공간 데이터 스트림을 공간 상황 스트림으로 변환하는 과정

서는 1) 센서 스트림을 실시간으로 전달받고 처리하고, 2) 상황 인지에 용이한 형태로 가공해야 한다. 이러한 기능을 지원하기 위해 SCONSTREAM은 다음의 여섯 가지 주요 모듈들로 구성된다.

공간 DSMS(Spatial DSMS) - 센서 네트워크로부터 전달 받은 공간 데이터 스트림은 크기가 무한하기 때문에 부분적으로 연속적으로 처리되어야 한다. 공간 DSMS 모듈은 부분적인 스트림 데이터를 제한된 크기의 슬라이딩 윈도우(Sliding window)에 일시적으로 저장하고 <표 2>의 공간 연산자들을 적용하여 실시간으로 연속 질의를 수행한다. 연속 질의 결과를 만족하는 스트림 데이터는 공간 상황 스트림으로 변환된다.

Sensor Network

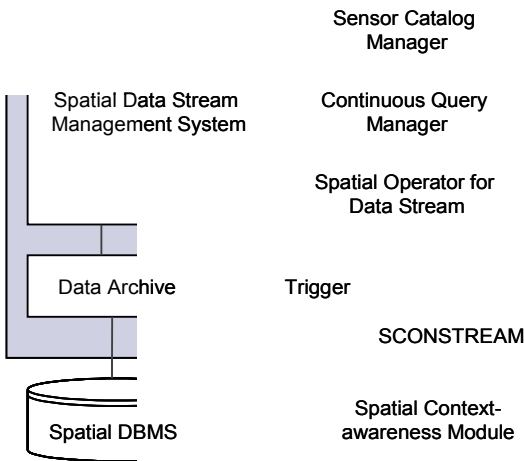


그림 4. SCONSTREAM의 구조

센서 카탈로그 관리 모듈(Sensor Catalog Manager) - 센서로부터 획득되는 스트림에는 x, y 좌표와 같이 원시적인 위치 정보만을 포함한다. 하지만 x, y 좌표 정보 보다 실제 객체가 위치한 공간에 대한 정보, 예를 들어, 객체 a는 현재 로비에 있다, 가 필요하다. 이러한 정보를 얻기 위해서 x, y 좌표 정보와 센서 메타정보 간의 결합(binding)이 필요하다. 센서 카탈로그 관리 모듈은 센서 네트워크로부터 전달되는 이종의 데이터 스트림의 카탈로그 정보를 관리한다. 센서의 카탈로그 정보는 데이터 스트림을 공간 상황 스트림으로 변환할 때 데이터 스트림과

해당 상황 간의 결합에 필요한 정보들을 제공한다. 그림 5는 센서 카탈로그 관리 모듈에서 관리하는 센서의 메타 정보를 웹 형태로 나타낸 것이다.

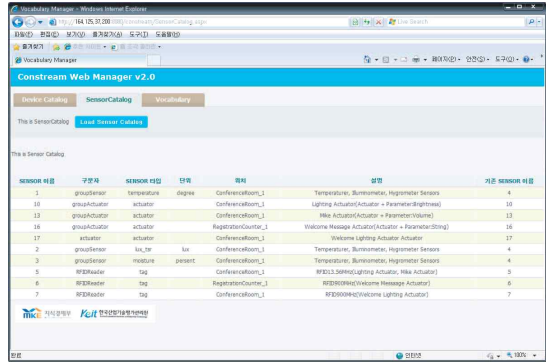


그림 5. 센서 카탈로그 정보

연속 질의 관리 모듈(Continuous Query Manager) - 공간적인 상황 조건은 데이터 스트림에 대해 연속적으로 수행된다. 연속 질의 관리 모듈은 공간적인 상황 조건에 대한 연속 질의 설정을 관리한다. 공간적인 상황 조건은 공간 연산자들의 집합으로 표현된다. [그림 6]은 상황 조건을 설정하기 위한 연속 질의 관리 모듈을 가시화한 것이다.

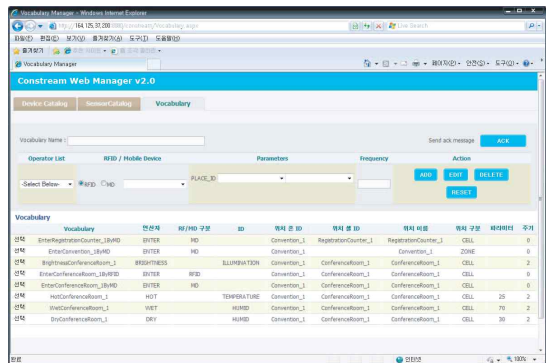


그림 6. 연속 질의 관리 모듈의 상황 조건 설정 화면

공간 연산자(Spatial Operator for Data Stream) - SCONSTREAM은 연속 질의를 정의하기 위해 <표 2>와 같은 공간적인 연산자들을 제공한다.

데이터 제공 모듈(Data Archive) - SCONSTREAM은 분석하고자 하는 상황을 만족하는 다양한 공간 상황 스트림이 생성한다. 하지만 생성되는 모든 공

간 상황 스트림이 상황에 필요하지 않을 수 있다. 따라서 데이터 제공 모듈은 생성된 공간 상황 스트림을 공간 DBMS(Spatial DBMS)에 저장하고, 공간 상황 인지 모듈이 필요한 공간 상황 스트림 정보를 전달한다.

트리거(Trigger) 모듈 - 센서로부터 전달되는 공간 데이터 스트림의 전달 주기는 높은 반면, 공간 상황 스트림이 전달되는 주기는 상대적으로 작다. 새로운 공간 상황 스트림이 생성되면, 그 정보가 공간 상황 인지 모듈로 트리거된다. 트리거를 위한 프로토콜은 [그림 7]에 설명되어 있다. 공간 상황 인지 모듈은 상황 인지를 위해 일부분의 공간 상황 스트림 정보만을 필요로 하는데, 보다 복잡한 상황 인지를 위해서 구체적인 공간 상황 스트림 정보를 데이터 제공 모듈을 통해 얻는다.

그림 4에 나타나 있듯이, 세 가지 외부 모듈이 SCONSTREAM과 연결되어 있다. 세 모듈은 센서 네트워크(Sensor network), 공간 상황 분석 인지 모듈(Spatial context-awareness module), 그리고 공간 DBMS(Spatial DBMS)이다. 본 논문에서는 다음의 세 가지 외부 모듈을 이용하여 SCONSTREAM을 구현하였다.

- 센서 네트워크 - COSMOS 미들웨어[8]
- 공간 상황 인지 모듈 - Top Quadrant[9]
- 공간 DBMS - Oracle Spatial[10]

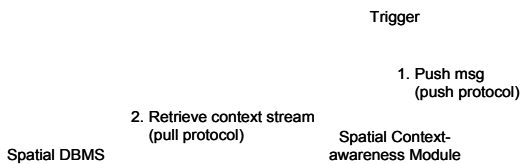


그림 7. 공간 상황 스트림을 제공하기 위한 두 가지 프로토콜

### 5. 성능 평가

본 장에서는 실험을 통해 앞서 구현한 SCONSTREAM의 성능을 평가한다. 실험을 위해 [그림 8]에 나타나있듯이 실제 서울의 컨벤션 건물의 한 층을 대상으로 500개의 이동 객체가 300초 동안 움직이는 위치 정보를 생성하였다. SCONSTREAM의 성능을 평가하기 위해 <표 3>과 같은 실제적인 시나리오를 정의하였다. 각 시나리오에 해당하는 공간 상황에 해당하는 공간 상황 스트림을 획득하기 위해서는 공간 연속 질의 설정이 매우 중요하다. <표 3>은 각 시나리오에 대한 연속 질의의 정보를 포함하고 있다.

SCONSTREAM의 성능에 영향을 끼치는 중요한 요소로 공간 데이터 스트림을 공간 상황 스트림으로 변환하는 과정에서 발생하는 데이터의 감소 비율(Reduction rate)이 매우 중요하다. 본 논문에서는 감소 비율을 다음과 같이 정의한다.

$$r = \frac{f_{CS}}{f_{DS}} (0 < r \leq 1)$$

$f_{CS}$  와  $f_{DS}$ 는 각각 공간 데이터 스트림과 공간 상황 스트림의 전달 속도를 나타낸다.  $r=1$ 의 경우, 공간 상황 스트림으로 변환 과정에서 데이터의 감소가 없음을 나타낸다. 즉, 센서로부터 전달받은 모든 데이터 스트림이 공간 상황 스트림으로 변환되었음을 나타낸다. 이 경우, 모든 센서 스트림이 공간 상황 인지 모듈로 전달되어 상황 인지 처리 부하를 크게 한다. 반면,  $r$  는 0 인 경우, 많은 데이터 스트림이 걸러지고 상대적으로 적은 공간 상황 스트림이 공간 상황 인지 모듈로 전달된다. 이 경우, 공간 상황 인지 모듈은 상황 인지를 실시간을 수행할 수 있다. 공간 상황에 대한 질의를 주의 깊게 디자인하여 SCONSTREAM의 감소 비율을 상당히 개선할 수 있다.

표 3. 성능 평가를 위한 시나리오와 그에 대한 연속 질의 정의

시나리오	연속질의
1. Auto transfer service - 해당 세션을 참석한 모든 참석자들의 컴퓨터에 세션의 발표 자료를 자동으로 제공한다.	1. Find all audiences who are INSIDE a conference room.
2. SMS service - 세션이 시작하면 세션 시작을 알리는 SMS를 세션 룸 밖에 있는 사람들의 모바일 폰으로 전송한다.	2. Find all audiences who are OUTSIDE a conference room.
3. Coupon service - 유티쿼터스 부스에 10초 이상 머무른 청중들에게 전자 점심 쿠폰을 그들의 모바일 폰으로 전송한다.	3. Find all audiences who STAY in the ubiquitous booth during 10 seconds.

표 4. 각 시나리오 별 감소율 및 평균 수행 시간

	시나리오1	시나리오2	시나리오3
감소율 (%)	0.0002	0.9142	0.0074
수행시간(ms)	80	192	8



그림 8. SCONSTREAM의 가시화 도구

<표 4>는 <표 3>의 각 시나리오에 대해 감소 비율과 질의의 평균 수행 시간을 측정된 것이다. 그 결과 시나리오 1, 3의 경우 감소 비율이 0.01 이하로 상당히 개선됨을 알 수 있다. 시나리오 2의 경우 감소 비율이 상대적으로 큰데(0.9142), 이는 시나리오 2의 경우 컨퍼런스 룸([그림 8]의 붉은 색) 밖에 존재하는 모든 이동 객체가 OUTSIDE라는 연속 질의 조건을 만족하기 때문이다. 여기서 중요한 것은 본 논문에서 제안한 SCONSTREAM이 센서로부터 획득되는 스트림 중 불필요한 정보를 걸러내고 오직 상황 조건을 만족하는 스트림만 상황 인지에 용이한 형태로 가공하여 전달함으로써 공간 상황 스트림에 집중되는 처리 부하를 줄일 수 있다는 것이다. 평균 수행 시간 역시 세 시나리오에서 실시간으로 처리할 수 있을 만큼 충분히 작음을 확인할 수 있다.

## 6. 결론 및 향후 연구

센서로부터 획득되는 공간 데이터 스트림을 처리하고 분석하는 것은 유비쿼터스 환경에서 공간 상황 인지를 현실화하는데 매우 중요한 요구 사항이다. 하지만 데이터 스트림 처리와 상황 인지 사이에는 중요한 기능적인 격차가 존재한다. 공간 데이터 스트림 처리 모듈은 센서 스트림을 실시간으로 처리하는 반면, 공간 상황 인지 모듈은 복잡한 상황 인지를 실시간으로 처리할 수 없다.

본 논문에서는 앞서 서술한 두 가지 모듈 사이에 존재하는 중요한 기능적인 격차를 해소하기 위해 SCONSTREAM이라는 공간 상황 스트림 처리 시스템을 제안하였다. 이 시스템의 중요한 기능을 요약하면 다음과 같다.

- 공간 데이터 스트림을 공간 상황 스트림으로 변환하고,
- 상황 데이터 스트림의 크기를 줄인다. 그리고,
- 공간 상황 스트림을 처리하기 위한 여러 시공간 연산자들을 제공한다.

본 논문의 기여로, 본 논문에서 제안하는 시스템은 센서로부터 획득되는 데이터 스트림으로부터 공간 상황 분석을 가능하게 하는 시초의 시스템이다. 향후 연구로는 시스템의 성능에 영향을 미치는 다양한 요소들을 조사하고 그것에 대한 성능을 분석하는 것이 있다. 그리고 본 논문에서 제안하는 SCONSTREAM을 기존의 STREAM과 Oracle Sensor Edge와 같은 시스템과 결합하는 것이 향후 연구에 포함된다.

## 참고 문헌

- [1] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Data, K. Ito, R. Motwani, U. Srivastava, J. Widom, "Stream: The standford data stream management system", IEEE Data Engineering Bulletin, Vol.26 No.1, 2003, pp. 19-26.
- [2] S. Chandrasekaran, O. Copper, A. Deshpande, M. J. Franklin, H. J. M, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, R. Reiss, M. Shah, "Telegraphcq: Continuous dataflow processing for an uncertain world", In Proceedings of the Conference on Innovative

- Data Systems Research, 2003, pp. 11-18.
- [3] J. Chen, D. J. DeWitt, F. Tian, Y. Wang, "Niagaracq: A scalable continuous query system for internet databases", ACM SIGMOD Record, Vol.29 No.2, 2000, pp. 379-390.
- [4] Oracle. Oracle sensor edge server. [http://www.oracle.com/technology/products/sensor\\_edge\\_server/index.html](http://www.oracle.com/technology/products/sensor_edge_server/index.html).
- [5] N. Tatbul, S. Zdonik, "Window-aware load shedding for aggregation queries over data streams", In Proceedings of the 32nd International Conference on Very Large Data Bases, 2006, pp. 799-810.
- [6] Y. D. Cai, D. Clutter, G. Pape, J. Han, M. Welge, L. Auvil, "Maids: Mining alarming incidents from data streams", In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, 2004, pp. 919-920.
- [7] J. Han, Y. Chen, G. Dong, J. Pei, B. W. Wah, J. Wang, Y. D. Cai, "Stream cube: An architecture for multi-dimensional analysis of data streams", Distributed and Parallel Databases, Vol.18 No.2, 2005, pp. 173-197.
- [8] M. Kim, J. W. Kee, Y. J. Lee, J.-C. Ryou, "Cosmos: A middleware for integrated data processing over heterogeneous sensor networks", ETRI Journal Vol.30 No.5, 2008, pp. 696-706.
- [9] T. Q. Korea. Top quadrant. <http://www.top-quadrant.co.kr/>
- [10] Oracle. Oracle spatial. <http://www.oracle.com/database/spatial.html>.
- [11] 김정준, 김판규, 김동오, 이기영, 한기준, "공간 DSMS 기반 RTLS의 설계 및 구현", 한국공간정보시스템학회 논문지, 제10권 제4호, 2008, pp. 47-58.
- [12] 심재민, 이용재, 주용완, 남광우, 류근호, "위치기반 푸쉬서비스 플랫폼 설계 및 구현", 한국공간정보시스템학회 논문지, 제11권 제4호, 2009, pp. 47-55.

논문접수 : 2010.03.04

수정일 : 1차 2010.04.22 / 2차 2010.04.26

심사완료 : 2010.04.28



### 권오제

2003년 부산대학교 컴퓨터공학과 졸업 (학사)

2005년 부산대학교 컴퓨터공학과 졸업 (석사)

2005년~현재 부산대학교 컴퓨터공학 (박사과정)  
관심분야 시공간DB, 시공간 데이터 스트림 처리, Ubiquitous Computing



### 김재훈

2008년 부산대학교 정보컴퓨터공학부 졸업(학사)

2010년 부산대학교 컴퓨터공학과 졸업 (석사)

관심분야 Ubiquitous Computing, 시공간DB



### 이기준

1984년 서울대학교 계산통계학과 졸업 (학사)

1986년 서울대학교 계산통계학과(전자계산학 전공) 졸업(석사)

1992년 프랑스 응용과학원 전자계산학과 전산학박사  
1993년~현재 부산대학교 정보컴퓨터공학부 교수  
관심분야 공간DB, 시공간DB, GIS, Telematics 및 Ubiquitous Computing, Indoor space