실시간 행위인식 시스템을 위한 스트림데이터 처리기 연산자 설계

정종영*, 이영구*, 한용구*
*경희대학교 컴퓨터공학과
{smashdown,yklee,ykhan}@khu.ac.kr

Design of Stream Data Processing Operators for Real-time Activity Recognition System

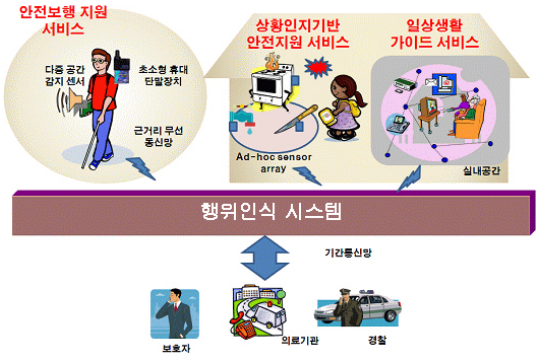
Jongyoung Jeong*, Young-Koo Lee*, Yong-koo Han*
*Dept of Computer Engineering, Kyung Hee University

요 약

본 논문은 오프라인 환경에서 연구되던 행위인식 연구들이 온라인 환경에서 통합되어 동작할 수 있도록 하기 위해 스트림데이터 처리기를 이용한 행위인식 시스템을 제안하였다. 스트림데이터 처리는 멀티 센서 환경에서의 방대한 센서 데이터를 처리하기 위한 기술로서, 행위인식 시스템의 실시간으로 발생하는 데이터를 즉각 처리하여 결과를 추론하여야 한다는 요구사항을 만족시킬 수 있다. 이를 위해 행위인식에 사용되는 필수적인 연산을 정의하여 스트림데이터 처리기에 추가할 수 있도록 행위 인식에 필요한 각종 연산자를 설계하였다.

1. 서론

유비쿼터스 기술이 고도화된 서비스를 제공하기 위해서는 지능적이며 적응적인 컴퓨팅을 지원할 수 있는 상황인지 기술이 필수이다[1]. 그 예로서 위치 인식, 지능형 오피스, 스마트 홈, 헬스케어 서비스를 들 수 있으며 이러한 기술은 모두 사용자의 현재 상태에 대한 정보를 바탕으로 사용자 행위를 추론하는 행위인식 기술을 기반으로 하고 있다. 그러나 현재까지 연구된 행위인식에 관련된 갖가지 연구는 제한된 범위 안에서 그들만의 특화된 기법을 이용하여 산발적으로 연구되고 있어 이에 따른 통합 및 확장을 위한 추가적인 연구가 필요한 실정이다. 그림1은 향후 발전할 다양한 행위인식 시스템이 통합되어 공존하는 유비쿼터스 환경을 나타내고 있다.



(그림 1) 행위인식 시스템을 이용한 다양한 서비스

스트림데이터 처리 기술[2]은 센서 환경 특히, RFID 기술을 운용함에 있어 발생하는 방대한 센서데이터를 처리하기 위한 프레임워크나 미들웨어로서 연구되어 왔다. 특히, 이기종의 많은 수의 센서데이터를 실시간으로 처리해야 한다는 관점에서 행위인식 기술을 실생활에 맞게 통합 및 확장하는데 유용할 것으로 판단된다. 하지만 행위인식 기술은 사용자 행위에 대한 최종적인 추론을 위해 크게 학습과 추론의 절차를 거치며 이를 위해서 레이블링과 전처리, 특징 추출 등의 다양한 연산과 기계학습의 복잡한 알고리즘까지를 포함하고 있는 특징을 갖고 있다. 이로 인해서 현재까지 연구된 스트림데이터 처리 기술이 관계형 데이터베이스에서 사용되는 관계대수 수준의 질의 서비스만을 제공한다는 점은 행위인식 관점에서의 스트림데이터 처리 기술의 한계로 작용한다.

이러한 문제점을 해결하기위해 본 논문에서는 행위 인식에 필요한 연산들을 기본 단위로 하여 고수준의 질의처리 서비스를 제공할 수 있는 스트림데이터 처리기를 제안하며 이를 위해 행위인식에 필요한 연산을 정의하고 그 구조와 연동 방식을 설계하였다.

2. 관련연구

2.1 행위인식 기술

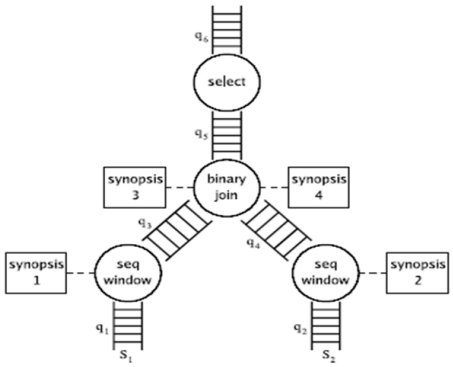
행위인식 관련 연구는 행위인식에 사용되는 센서의 종류와 추론알고리즘의 종류별로 다양한 방식으로 연구되었다. 가속도 센서를 이용하여 신체 이동의 패턴을 분류한 연구[3]가 있고, RFID를 이용하여 사용자가 사용하는 사물을 기반으로 행위를 추론하는 연구가 있으며[4], 비디오를 이용한 연구도 있다.[5] 또한, 행위인식을 위해서 필수적으로 사용되는 기계학습 알고리즘에 대해서 다양한 접근법을 제시하는 연구들도 함께 수행되어 왔다. 하지만 대용량 스

트림 형태의 센서 데이터들이 행위 데이터베이스에 저장되어 있는 가정 하에 연구되었기 때문에 행위인식의 정확성 향상에 커다란 공헌을 하였지만 센서들이 끊임없이 발생하는 대용량 센서 데이터 환경을 고려하지 않고 있다.

그러므로 본 연구는 기존의 연구된 행위인식 기법들이 대용량 스트림 센서 데이터 환경에서 효율적으로 동작할 수 있는 행위인식 스트림데이터 처리기를 설계하였다.

2.2 데이터스트림 처리기술

스트림데이터 처리 기술은 센서로부터 생성되는 데이터는 실시간으로 나열되어 스트림의 형태를 이루게 된다는 점에서 시작되었다. 이에 따라 연속적인 질의 처리를 위한 스트림데이터 처리기술이 등장하게 되었으며 대표적인 스트림데이터 처리 기술로는 스탠포트 대학의 STREAM[6], 위스콘신 매디슨 대학의 NiagaraCQ[7] 그리고 브라운 대학과 M.I.T의 Aurora[8]등이 있다. 기존의 스트림데이터 처리 연구는 철저하게 데이터 관점에서 실시간으로 발생하는 저수준의 센서데이터를 관리하고 질의하기 위한 방법에 집중하고 있기 때문에 행위인식을 위한 고수준의 연산에 대한 질의를 서비스해주지 못한다는 한계를 지니고 있으며 현재까지 연구된 대부분의 스트림데이터 처리 연구는 저수준의 센서데이터에 대해서 관계형 데이터베이스 수준의 질의 서비스만을 제공하는데 그에 해당하는 연산으로는 selection, projection, join, sort, 등이 있다. 나아가 센서 데이터가 방대하며 실시간으로 발생한다는 특징을 가지므로, 이를 해결하기 위한 저장소 및 메모리 관리 기법과 스케줄링 기법 등을 다루고 있으며, 추가로 기존의 이미 존재하는 데이터에 대한 수동적인 질의 서비스가 아닌 실시간으로 발생하는 데이터에 대한 능동적인 질의 서비스를 다루고 있다.



(그림 2) STREAM에서의 연산자 배치 구조[6]

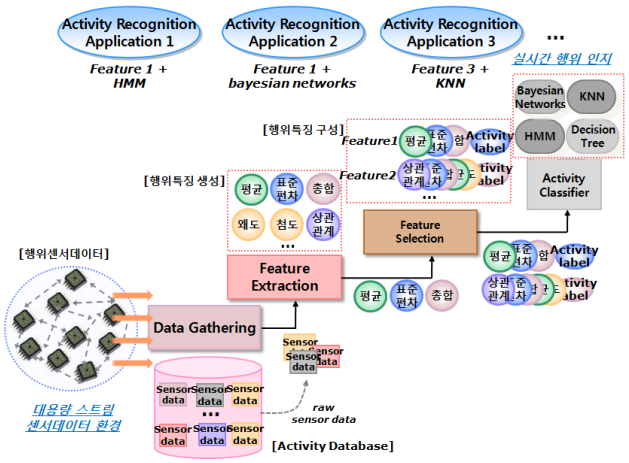
그림2는 스탠포트대학의 STREAM의 질의처리 과정을 도식화 한 것이다. STREAEM에서는 질의처리 연산을 기본 단위로 하여 스케줄링 / 배치 한 후 동시 다발적으로 처리해 가는 방식을 취하고 있다. STREAM 뿐만 아니라 M.I.T의 Aurora 및 위스콘신 매디슨 대학의 NiagaraCQ 등, 그 밖의 다른 대부분의 스트림데이터 처리 기술에서도 기본적인 연산은 기초적인 질의처리 연산에 제한되고 있으며 이를 활용하여 처리하는 기능은 상위 애플리케이션이 책임지고 있다. 또한, 데이터 관점에서 실시간으로 발생하는 저수준의 센서데이터를 관리하고 질의하기 위한 방법에 집중하고 있기 때문에 행위인식을 위한 고수준의 연산에 대한 질의를 서비스해주지 못한다는 한계를 지니고 있다.

3. 스트림데이터 기반 행위인지 시스템 설계

본 연구는 대용량 스트림 형태의 센서데이터 환경에서 행위를 실시간으로 인지하기 위하여 행위 인지의 각 수행 절차를 기본연산으로 하는 스트림데이터 처리기 기반의 행위인지 시스템을 설계하였다.

3.1 행위인지 스트림 처리기의 기본 연산 단위

하나의 행위인식 시스템은 다양한 애플리케이션에서 사용이 가능하다. 각 애플리케이션은 그림 4에서와 같이 목적에 따라 각기 다른 센서 데이터, 행위 특징 및 분류기를 사용할 수 있다. 이는 센서데이터의 복합적인 연산을 기초로 복잡한 알고리즘들이 사용되는 행위인식 시스템이 동일한 수행 절차를 중복처리하게 되어 시스템의 성능을 저하시키는 원인이 된다. 그림3은 행위인식의 필수적인 절차를 묘사하고 있다.



(그림 3) 행위인지의 정형적인 수행 절차

위 그림의 공통적인 절차에 대해서 각 연구는 고유의 복잡한 알고리즘 및 기법을 마련하여 해결하고 있으며 그 방식은 저수준의 센서데이터에 대한 복합적인 연산을 기초로 하고 있다. 향후 실생활에서 기능해야 할 실용적인 행위인식 시스템은 갖가지 센서를 통합하여 다양한 행위 레벨에 따른 복합적인 행위인식이 가능해야 한다고 할 때, 위의 공통적인 절차에 대한 다수의 중복이 큰 문제가 될 수 있다. 그러므로 본 논문에서는 위 그림에서의 각 연산을 기본 단위로 하며 그 질의 서비스까지 제공 가능한 스트림데이터의 연산을 정의하며 이를 이용한 행위인식 시

스템을 설계한다.

3.2 행위인식 연산의 설계

행위인식의 필수적인 절차에 대한 연산의 범주를 살펴보면 크게 그림4 에서와 같다. 그 중 행위인식에 사용되는 필수 연산은 대부분 특징 추출과 패턴분류 / 기계학습에 관련된 연산이며 그 연산을 살펴보면 아래와 같다.

(1) 전처리 연산

수집된 센서 데이터에 대해 행위 인식에서 사용되어질 형태로 가공 및 필터링 하는 연산으로 구성된다. 저수준의 센서데이터에 대해서 값의 타입과 범위를 제한하는 필터링 연산과 각 센서에서 발생하는 데이터의 복합구조에 대한 분해 및 조립을 지원하는 파싱연산이 있다. 그 상세는 아래와 같으며 Pi는 처리되어야 할 센서데이터 튜플, T는 한계값, S는 출력 스트림이다.

- Filter(T, P1, . . . , Pm)(S)
- Normalize(T, P1, . . . , Pm)(S)

(2) 특징 추출 연산

기본적으로 평균, 표준편차, 연관 연산이 있으며 추가로 센서데이터의 주기적 특징을 분류하기 위해 퓨리에 변환과 라플라스 연산이 있다. 그 외에도 센서데이터의 통계적인 추론을 위해서 기본적인 확률 연산과 베이시안 모델 연산 등 신호처리 관련 연산이 있다. Si는 입력 스트림이고 S는 출력스트림이다. 특징추출연산들은 각종 센서데이터 스트림을 입력으로 받아 단위 특징을 추출하 단일 스트림으로 내보내게 되어있다.

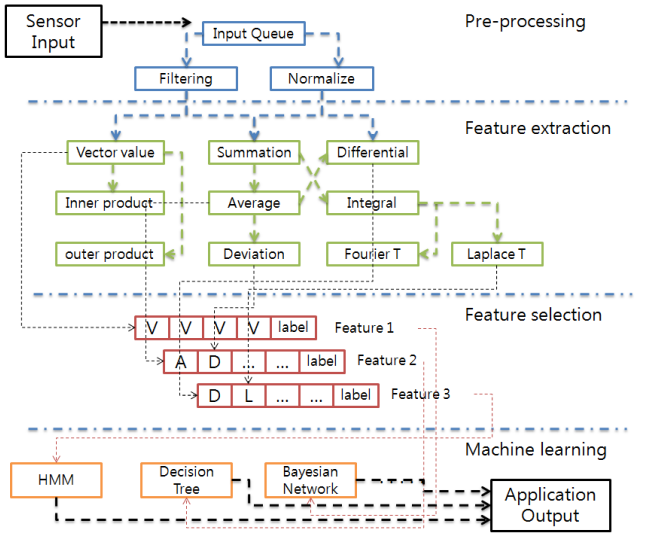
- Average(S1, . . . Sm)(S)
- StandardDeviation(S1, . . . Sm)(S)
- Correlation(S1, . . . Sm)(S)
- FFT(S1, . . . Sm)(S)
- LT(S1, . . . Sm)(S)

(3) 패턴 분류 / 기계학습 연산

다양한 기계학습 알고리즘이 행위인식을 위해 사용되었는데 이를 단위연산으로 캡슐화 한 것이다. 대표적으로 사용되는 기계학습 알고리즘으로 Decision Tree, Bayesian Network, Support Vector Machine, NNH 등이 있다. 각 알고리즘은 객체지향적으로 설계되었으며 각각 학습연산과 추론 연산을 포함하고 있으며 그 명세는 그림5에 나타나 있다.

전처리 단계, 특징추출단계, 특징선택, 기계학습 단계에 해당하는 연산들에 대해서 본 연구에서는 3가지 애플리케이션을 가정하고 각 애플리케이션에서 요구하는 행위인식 결과를 생성하기 위해 필요한 연산들이 정의되었다. 현재 연구되고 있는 다양한 행위인식 애플리케이션을 지원하기 위해서는 보다 많은 연산들이 필요하며 특정 연구에서는 그들만의 새로운 연산을 통해서 특징을 추출하기도 하므로 이를 반영하기 위한 설계 구조 또한 필요하다.

행위인식 연산의 수행 구조를 살펴보면 그림4와 같다. 행위인식에 필수적인 절차 4가지에 대해서 각 절차에 해당하는 연산들이 존재하고 각 연산들은 상호연관성의 유무에 따라 연산 절차가 공유된다. 예를 들어 표준편차를 구하는 연산은 평균을 구하는 연산의 결과값을 이용하고 평균을 구하는 연산은 총합을 구하는 연산을 이용하는 것이다. 이를 통해서 각 연산이 갖게 되는 비용을 절감하는 효과를 거둘 수 있다.



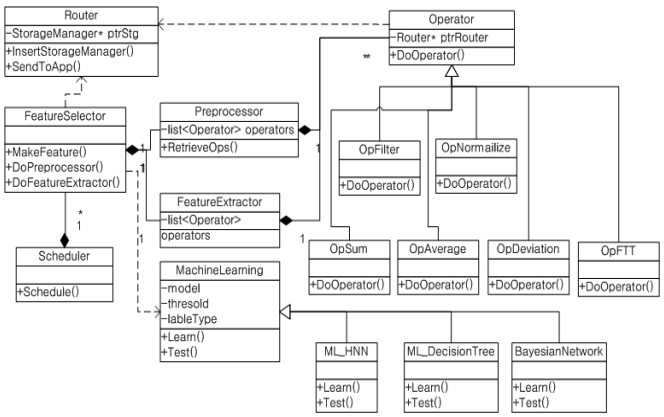
(그림 4) 행위인식 관련 연산 집합의 구조

세 가지 행위인식 애플리케이션을 가정할 때, 입력되는 센서데이터에 대해서 전처리 단계에서는 필터링과 정규화를 거치게 되고 특징추출단계에서는 벡터관련 연산과 통계관련 연산, 주파수 영역 관계연산들이 사용된다. 각 연산의 결과로 생성되는 특징들은 특징선택 단계에서 조합되어 레이블링이 붙여져 하나의 특징벡터를 생성하게 된다. 생성된 특징벡터는 기계학습의 입력값으로 사용되게 되고 기계학습을 통해 추론된 결과가 상위 애플리케이션에 전달되어 행위인식에 사용되게 된다.

그 절차를 설계하여 클래스다이어그램으로 표현하면 그림5와 같다. 라우터와 스케줄링 클래스는 기존 스트림데이터 처리기에 존재하는 클래스로 라우터는 센서데이터 및 연산결과값의 입출력을 처리하고 스케줄러는 애플리케이션의 질의 요청에 맞추어 각 연산들을 배치 및 수행하는 역할을 한다.

행위인식에 사용되는 각 연산은 추상클래스 Operator를 상속받는 Strategy 패턴을 적용하였다. 이 구조를 통해서 추가되는 애플리케이션에 대해 사용자 정의 연산을 추가할 수 있는 편의성을 확보할 수 있다. 각 연산은 크게 전처리 연산과 특징추출 연산으로 분류될 수 있는데 전처리기에서는 해당 애플리케이션의 요구에 맞는 전처리 연산을 리스트로 포함하며 특징추출기에서도 마찬가지로 특징추출 연산을 포함한다. 특징선택기는 전처리기와 특징추출

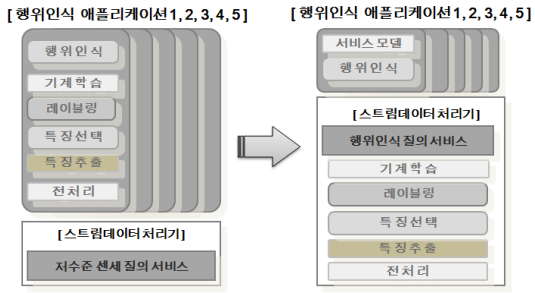
기를 포함하고 있으며 전처리와 특징추출절차를 거쳐 생성되는 단위 특징들을 조합하여 특징벡터를 생성하게 된다. 생성된 특징벡터는 기계학습 연산에 사용되는데, 기계학습연산 또한 필요에 따라 사용자 정의가 가능하도록 Strategy패턴을 적용하였으며 기계학습의 결과로 생성되는 추론값이 상위 애플리케이션에 전달되어 사용되게 된다.



(그림 5) 행위인식 연산 구조 클래스 다이어그램

6. 결론 및 향후 연구

본 논문에서는 행위인식 기술이 실생활에 적용되기 위해서 발생할 수 있는 문제점을 살펴보고 이를 해결하기 위한방법으로 스트림데이터 처리 기술을 도입하였다. 이를 위해서 행위인식을 지원하기 위해 스트림데이터 처리 기술에 포함되어야 하는 연산을 정의하고 기타 관리 기능을 설계하였다. 그 결과 그림6에서 볼 수 있는 것처럼 기존의 행위 인식 애플리케이션들은 중복적으로 각자의 행위 인식 절차를 포함하던 것에 반해서 행위인식에 필수적인 절차들을 내포하고 있는 스트림데이터 처리기를 이용하는 방법으로 기존 방식에 비해서 보다 추상화되고 간편한 애플리케이션 구현이 가능하도록 하였다.



(그림 6) 행위인식 연산을 포함한 스트림데이터 처리기

향후에는 본 논문에서 제안하는 시스템을 구현하고 깊게 언급하지 못한 다양한 레이블링 기법과 본 논문에서 언급하지 못한 추가적인 특징추출 알고리즘 등을 구현하여 보다 강화된 행위인식 시스템을 구현할 수 있도록 할 계획이다.

Acknowledgement

본 연구는 한국과학재단을 통해 과학기술부의 국가지정연구실사업(NRL, National Research Lab)으로부터 지원받아 수행되었습니다(R0A20070002010112008).

참고문헌

[1] 임신영, 허재두, "상황인식 컴퓨팅 응용 기술 동향", 전자통신동향분석 제19권 제5호. 2004

[2] Daniel J. Abadi1, Don Carney, U˘gur C¸ etintemel, Mitch Cherniack1, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, Stan Zdonik , "Aurora: a new model and architecture for data stream management", The VLDB Journal, 2003

[3] Ling Bao and Stephen S. Intille, "Activity Recognition from User-Annotated Acceleration Data", Pervasive 2004 (21-23 April 2004), pp. 1-17.

[4] Emmanuel Munguia Tapia, Stephen Intille, Kent Larson, "Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors", pp. 158 - 175 in Proceedings of PERVASIVE 2004

[5] Neil Robertson, Ian Reid, "A General Method for Human Activity Recognition in Video", Journal of Computer Vision and Image Understanding, Vol. 104, No. 2 pp. 232-248, 2006

[6] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava and J. Widom, "Stream: The Stanford Data Stream Management System" , in IEEE Data Engineering Bulletin, vol. 4(1), 2003

[7] S. Chandrasekaran, O. Copper, A. Deshpande, M. J. Franklin, H. Joseph M, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss and M. Shah, "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World", in Proceedings of the Conference on Innovative Data Systems Research, pp.11-18, 2003

[8] Jianjun Chen, David J. DeWitt, Feng Tian and Yuan Wang, "NiagaraCQ: A Scalable Continuous Query Stream for Internet Databases" , in ACM SIGMOD Record, pp.379-390, 2000