

센서 네트워크하에서의 효율적 물체 추적 알고리즘 개발

Development of the Effective Motion Tracking Algorithm Under Sensor Network

김시환*, 김성호**

* 군산대학교 전자정보 공학부

E-mail: sihwan@gmail.com

** 군산대학교 전자정보 공학부

E-mail: shkim@kunsan.ac.kr

2

요 약

본 연구에서는 한정된 전원으로 구동되는 센서 네트워크 환경 하에서 물체의 이동을 검출하고 예측을 통해 효과적인 추적을 가능케 함으로써 missing-rate를 최소로 하는 새로운 형태의 알고리즘을 제안하고 시뮬레이션을 통해 제안된 방법의 유용성을 입증하고자 한다. 제안된 기법에서는 물체의 이동과 관련된 센서 노드들로부터의 정보 및 이를 기반으로 센서 노드에 장착된 A/D변환기의 임계값을 적응적으로 변화시킴으로써 물체의 missing-rate를 최소화 시키고자 하였다.

Key Words : Sensor Network, Object Tracking

1. 서 론

최근 들어 물리적 공간과 전자적 공간을 융합한 유비쿼터스란 개념이 널리 사용되고 있다. 유비쿼터스 공간이란 모든 물리공간에 컴퓨터를 집어넣음으로써 사람과 컴퓨터뿐만 아니라 사람과 사물, 사물과 사물이 모두 유 무선으로 연결되어 서로 대화하고 알아서 스스로의 행동을 결정할 수 있는 환경을 의미한다. 이러한 유비쿼터스 환경의 구체적 구현으로, 최근 급속히 발전되고 있는 무선통신기술과 반도체 설계 기술을 이용한 저가격, 저 전력의 다양한 센서노드들에 기반한 센서 네트워크를 들 수 있다. 센서네트워크는 기본적으로 계측 및 무선통신 기능이 탑재된 다수의 센서노드들로 구성된다. 이러한 센서노드들은 자체적으로 네트워크를 형성하는 것이 가능하며 이로 인해 측정된 정보를 자체적으로 형성된 망을 통해 인터넷과 같은 기간망으로 전달할 수 있어 광범위한 지역의 생태 모니터링, 지진 감시 및 군사용 등에 폭넓게 적용되고 있다.

물체의 위치 추적은 현대 사회가 유비쿼터스 환경으로 가는데 있어 반드시 해결해야 할 문제 중 하나이며 특히 센서 네트워크상에서의 물체추적은 어플리케이션의 특성상 단순한 환경모니터링 어플리케이션과는 전혀 다른 영역이다. 센서 네트워크에서 이동성을 갖는 물체

를 정확히 추적하기 위해서는 네트워크를 구성하고 있는 많은 노드들이 지속적으로 서로 협업하여야 한다. 그러나 센서 네트워크는 한정된 전원으로 최대한 오래 사용할 수 있어야 하며 따라서 이동성을 갖는 물체의 추적에는 최소한의 missing-rate를 유지하면서 최소한의 에너지를 사용해야만 한다. 최근 이와 관련하여 많은 연구들이 진행되고 있으며 크게 분류하면 첫째, 물체에 근접한 하나의 노드만 사용하여 그 노드의 위치로 물체의 위치를 근사적으로 나타내는 방식과 둘째, 물체에 근접한 여러 개의 노드들로 동적 클러스터를 구성하여 나가는 방식, 셋째, 물체의 다음 위치를 예측하고 물체의 이동경로 주변의 노드를 이용하여 물체를 모니터링 하는 방식들을 들 수 있다.

본 연구에서는 한정된 전원으로 구동되는 센서 네트워크 환경 하에서 물체의 이동을 검출하고 예측을 통해 효과적인 추적을 가능케 함으로써 missing-rate를 최소로 하는 새로운 형태의 알고리즘을 제안하고 시뮬레이션을 통해 제안된 방법의 유용성을 입증하고자 한다. 제안된 기법에서는 물체의 이동과 관련된 센서 노드들로부터의 정보 및 이를 기반으로 센서 노드에 장착된 A/D변환기의 임계값을 적응적으로 변화시킴으로써 물체의 missing-rate를 최소화 시키고자 하였다. 논문의 구성은 다음

과 같다. 2장에서는 기존의 센서 네트워크상에서의 물체 움직임 검출 기법 등에 대해 간단히 설명한 후, 3장에서는 본 논문에서 제안된 기법에 대해 설명하고, 4장에서는 제안된 기법의 유용성 확인을 위해 가정된 상황 하에서의 시뮬레이션을 수행하며 5장에서는 결론을 기술한다.

2. 물체추적 관련 연구 동향

기존의 센서네트워크에서 물체추적에 대한 연구들은 크게 Naive방식, 계획적 모니터링 방식, 지속적 모니터링 방식, 동적 클러스터링 방식, 예측기반의 방식 등으로 분류할 수 있다. 그 중에서 지속적 모니터링, 동적 클러스터링, 예측기반. 등의 방식은 기존의 센서네트워크에서의 일반적인 응용에서 부가적으로 지원되는 Naive, 계획적 모니터링 방식과 달리 전문적으로 물체추적을 위해 연구되었다. 각각을 살펴보면 다음과 같다.

Naive 방식은 모든 센서노드들은 활성화 상태를 계속 유지하면서 감지된 결과를 주기적으로 base station에 보내주기 때문에 에너지 소모가 너무 많다.

계획적 모니터링 방식은 모든 센서노드들과 base station이 잘 동기화되었다고 가정하고 sleep과 wake up 두 개 상태사이에서 transit할 수 있다. 센서노드는 자신이 센싱 해야 할 순간이 되면 바로 센싱 하고 데이터를 base station에 보낸다. 모든 센서노드들은 X초만큼 활성화 상태에 있다가 (T-X)초 동안 수면상태에 들어간다. 하지만 모든 센서노드들과 base station이 잘 동기화 되었다는 가정은 현실적으로 실현이 어렵다. 또한 물체를 추적하기 위해 사용되는 센서노드의 수량이 많으므로 에너지 소모가 많게 된다.

지속적 모니터링 방식은 모든 센서노드들이 주기적으로 깨어나 센싱 하다가 물체가 자신의 센싱 범위에 들어온 센서노드만 활성화 된다. 활성화노드는 물체가 이웃 노드의 센싱 범위에 들어가기 직전에 이웃 노드를 깨워준다. 하나의 노드로만 모니터링하기 때문에 에 missing-rate가 높으며 활성화 노드는 계속 센싱 하고 있어야 하기 때문에 에너지 소모가 불균형적이며 전반적인 네트워크의 생명주기에 영향을 준다[1].

동적 클러스터링 방식은 파워가 강한 CH노드와 파워가 약한 SN 노드로 구성된다. 물체를 센싱한 신호가 제일 강한 CH노드가 클러스터의 헤드가 되고 주변의 SN 노드들로 클러스터를 구성한다. SN.노드들은 센싱한 정보를 클러스터 헤드인 CH노드에 전송해주고 CH노드

는 수집한 정보들을 base station에 전달해 준다. 물체를 놓치지 않기 위해서 모든 CH노드들은 지속적으로 센싱을 해야 한다. 오직 CH노드들만 클러스터 헤드가 될 수 있기 때문에 CH노드가 균일하게 뿌려지지 않은 지역에서는 물체의 추적이 어렵게 된다. 그리고 클러스터가 일단 구성된 후에는 헤드를 Rotation하지 않기 때문에 물체의 이동이 매우 느린 경우에 특정한 클러스터 헤드의 에너지 소모는 많아진다. 따라서 CH노드들 사이의 에너지 소모가 불균형해지고 전반적인 네트워크의 생명주기에 영향을 준다.

예측기반의 방식은 예측모델을 통하여 물체의 다음 위치를 예측하면 Wake up Mechanism을 통하여 물체가 현재 노드의 센싱 범위를 벗어나 인접노드로 들어가기 직전 인접노드를 깨우며 물체를 놓쳐버리면 전체 네트워크 범위에서 물체 추적을 복구한다.

Application의 부동한 요구에 따라 아래의 Wake up Mechanism 중 하나를 선택한다.

- 목표위치 부근에서 하나의 노드만을 선택하여 깨운다.
- 현재위치에서 목표위치까지의 이동경로 상에 있는 노드들을 모두 깨운다.
- 현재위치에서 목표위치까지의 이동경로 주변의 모든 노드들을 깨운다.

선택된 Wake Up Mechanism이 다름에 따라 존재하는 문제점 역시 다르다. 먼저, 목표위치 부근에서 하나의 노드만을 선택하여 깨울 경우 위치예측이 완전히 정확한 상황에서만 효과적이지만 반면에 하나의 노드만 계속 센싱하고 있어야 하기 때문에 에너지 소모가 불균형적이다. 또한 위치예측의 100% 정확성을 보존하기 어려운 현실에서는 missing-rate가 상당히 높기 때문에 매번 추적을 복구하기 위해서 들어가는 부하가 엄청나게 크다. 둘째, 현재위치에서 목표위치까지의 이동경로 상에 있는 노드들을 모두 깨울 경우, 위치추정의 약간의 오차를 허용하지만, 이동경로상의 노드들만 계속 모니터링하기 때문에 물체가 도중에 이동 방향을 바꾼다면 missing-rate가 높게 되고 추적을 복구하는데 드는 부하도 많다. 또한 이동경로상의 노드들은 모두 깨워 동시에 센싱하기 때문에 에너지 소모방면에서 비효율적이다. 셋째, 현재위치에서 목표위치까지의 이동경로 주변의 모든 노드들을 깨울 경우, 위치추정의 일정한 오차를 허용하지만 이동경로 주변의 모든 노드들을 깨워 동시에 센싱하기 때문에 missing-rate가 낮은 반면에 에너지 낭비가 많게 된다[2-4].

본 논문에서는 기존의 연구와는 달리 물체의 이동 경로를 방향을 따라 센싱 임계값을 능동적으로 조절하여 이동 물체에 대한 missing-rate를 최소화하는 것을 목적으로 한다.

3. 제안된 물체 추적 기법

일반적으로 센서 네트워크상에서 이동하는 물체의 추적을 위해서는 마이크 및 조도 등과 같은 다양한 형태의 센서 등이 사용되고 있으며 물체의 존재는 센서 출력이 설정된 임계값을 넘을 경우에 인식된다. 따라서 각각의 센서 노드에 탑재되는 센서 임계치의 최적 설정은 물체의 인식 및 경로의 추적 성능에 큰 영향을 미치게 된다. 본 연구에서는 이러한 점에 착안하여 센서의 임계치 설정을 적응적이게 함으로써 물체 추적시의 missing-rate를 최소화시킬 수 있는 새로운 형태의 알고리즘을 제안하고자 한다.

전체 알고리즘은 다음과 같이 이동 감지 과정, 이동 감지 후 과정, 측정 메시지 수신 과정의 세 부분으로 나눌 수 있다.

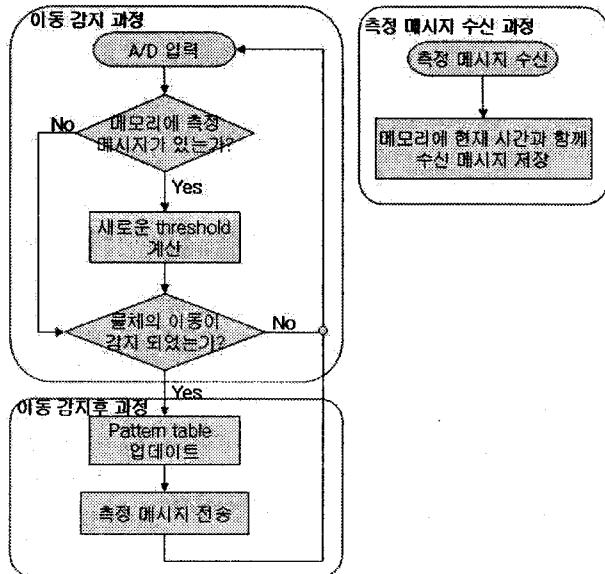


그림 1. 제안된 물체 추적 알고리즘

이동 감지 과정은 센서로부터 들어온 A/D값을 사용하여 물체의 이동을 감지하는 과정이다. 이때 pattern table을 바탕으로 새로운 임계값을 계산하는 과정이 중요한데 이것은 3.4절에서 설명하도록 하겠다. 이동 감지 후 과정은 패턴 테이블을 업데이트하고 물체 이동 감지에 관한 측정 메시지를 주변노드로 송신하는 과정이다. 주변 노드는 측정 메시지를 수신하게 되

면 이동 감지 과정과 패턴 테이블을 업데이트 하는데 쓸 수 있도록 메모리에 기록한다.

3.1 이동 감지 과정

이동 물체의 추적은 물체의 이동이 감지된 센서노드로부터 시작된다. 즉, 어떤 센서노드 근방에서 물체의 이동이 감지되면(센서 값이 설정된 임계값을 초과할 경우) 이 센서 노드는 자신의 주변에 표 1과 같은 패킷 메시지를 broadcast 한다.

표 1. broadcasting 되는 패킷 메시지 구조

Field	Description
address	target address (0xFFFF)
type	AM message type
group	group address
length	data length
sender ID	unique identifier of sending node
precision	sensing precision
crc	crc

sender ID는 물체를 감지한 센서노드의 고유 ID를 의미하며 precision은 물체 감지에 대한 정확도를 나타낸다. 센서로부터 측정된 A/D값이 설정된 임계값보다 높은 값일수록 precision 값도 높아진다. 즉, 이 값이 클수록 물체를 확실히 감지했음을 의미한다. precision값을 만드는 식은 식 1과 같다.

$$P = (V_{ad} - T)^2 \quad (1)$$

여기서 V_{ad} 는 A/D 변환을 통해 얻어진 센서 값이고 T 는 해당 센서의 임계값이다. P 는 (1)의 결과로 나온 precision 값이다. precision이 사용되는 이유는 정확하지 않은 측정 결과 때문에 주변의 노드의 측정에 영향을 줄 수 있기 때문이다. 표 1의 메시지를 받은 주변 노드들은 메시지를 받은 시각과 함께 자신의 메모리에 메시지를 저장해 놓는다. 이 저장된 메시지는 일정 유효시간이 경과하면 자동적으로 소멸된다. 설명을 위해 4개의 센서노드가 존재할 경우, 각 센서 노드의 메모리에 저장되는 내용과 패턴 테이블 내용을 나타내면 그림 2와 같다. 그림에서 2번 센서 노드는 1번 센서 노드에 대해 높은 가산점을 주었다. 이 값은 다음에 1번 센서 노드로부터 측정 메시지가 전달될 때 자신의 측정 범위가 물체의 다음 경로가 될 것이라 예측하고 임계값을 낮추는 역할을 한다. 이러한 기법의 기본 아이디어는 조건 반

사에서 나왔다고 할 수 있다. 우리는 음식을 먹기 전에 미리 위산이 나와 소화가 용이하도록 돕는다. 마찬가지로 본 논문에서 사용한 기법 또한 물체 이동 경로를 미리 예상하고 임계값을 낮추어 물체 이동 감지를 확실하게 한다.

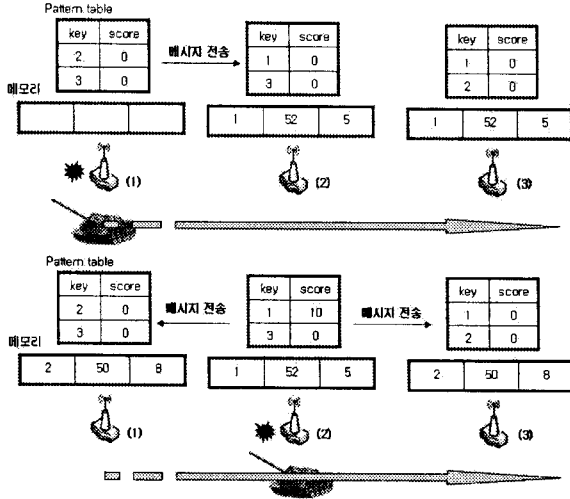


그림 2. 3개의 센서노드가 존재할 경우 각 노드에 저장되는 내용

3.2 이동 감지 후 과정

물체의 이동이 감지되면 다음 감지 과정을 위해 현재 패턴 테이블을 업데이트 시키고 주변 노드로 측정 메시지를 broadcast 한다.

표 2. 패턴 테이블의 데이터 형 구조

Field	Description
score	score of node
time	average time

표 2 는 패턴 테이블의 구조를 나타내며 여기서, score는 가산점 이고 time은 센서 노드의 측정 메시지를 수신 받은 후 물체 이동을 감지할 때까지의 평균 시간이다. 패턴 테이블을 업데이트 시키는 방법은 테이블에서 패턴으로 인정된 노드, 즉 물체 이동이 감지되었을 때 메모리에 저장되어 있는 센서 노드에 가산점을 부여 한다. 또한 패턴으로 인정되지 못한 모든 주변 노드의 가산점을 줄인다. 가산점을 늘리는 비율과 줄이는 비율은 경험적으로 5:1에서 10:1을 사용하는 것이 안정적 이다. 가산점 이외에도 평균 경과 시간을 업데이트 시켜야 하는데 평균 경과 시간의 업데이트 방법은 식 2 와 같다.

$$i_{(k)} = \frac{3i_{(k-1)} + t_{(k)}}{2} \quad (2)$$

여기서 i 는 평균 경과 시간이고 t 는 물체가동이 감지된 현재 시간에서 메모리에 있는 도착 시간을 뺀 시간이다. 패턴 테이블을 모두 업데이트 시킨 후 표 1 과 같은 측정 메시지를 주변 노드로 broadcast 한다. broadcast 한 메시지를 받은 노드는 3.3 절의 측정 메시지 수신 과정에 들어간다.

3.3 측정 메시지 수신 과정

무선 RF 를 통해 수신된 데이터는 broadcast 된 메시지이기 때문에 아직 의미를 가지는 값이 아니다. 때문에 임시적인 저장소인 메모리에 표 3 와 같은 구조로 저장된다.

표 3. 메모리 데이터 구조

Field	Description
address	address of sends node
precision	precision of message
time	received time

address 와 precision 은 패킷에서 추출한 값이고 time 은 현재 시간을 사용한다. 이 time 과 A/D 과정을 통해 물체를 감지한 시간까지가 (2)의 경과 시간이 된다.

3.4 새로운 임계값 계산

측정 메시지 수신 과정 후 A/D값을 가지고 물체의 감지 유무를 판단하는 임계값을 새로 설정하게 된다. 이것을 의사코드로 표현하면 아래와 같다.

```

tval = progress time / expected time
tval = If tval < 1 then 1
temp = (init score / current score) * tval
result = If temp > 1 then init threshold
        else init threshold * temp
    
```

여기서 progress time 은 현재 시간에서 표 3 의 time 을 뺀 값이고 expected time 은 표 2 패턴 테이블의 time 이다. 또한 init score 는 초기 가산점 이고 current score 는 패턴 테이블에서의 현재 score 이다.

4. 시뮬레이션 및 고찰

본 연구에서는 시뮬레이션을 위해 TOSSIM 을 사용하였다. TOSSIM은 TinyOS 기반의 센서 네트워크에서 TinyOS의 동작을 검증하고, 응용프로그램의 개발을 도와주는 시뮬레이터이다. 그림 3과 같이 50m X 50m 구간에 20개의 센서 노드를 그리드 형태로 배치하였다.

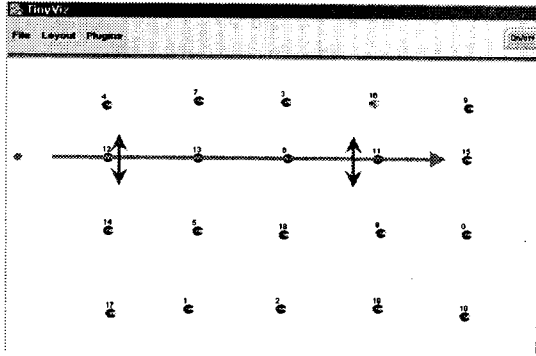


그림 3. TOSSIM 시뮬레이션 환경

이동 물체는 y축 방향으로 gaussian random 하게 진동하면서 x축 방향으로 진행하며 각각의 센서 노드의 센서에 다음과 같이 표현되는 영향을 준다.

$$P_r = \frac{P_o}{1 + e^{d*0.5}} \quad (3)$$

P_o 는 물체가 가지고 있는 초기 센서 값으로 실험을 위한 상수이다. e 는 자연로그이고 d 는 물체로부터 센서 노드까지의 거리이다. x축 방향으로 진행하다가 끝에 도달하면 시작 부분으로 이동하여 다시 x축 방향으로 진행한다. 이러한 실험으로 일정한 임계값을 가지고 이동 물체를 감지하는 방식과 본 연구에서 제안된 기법과의 비교를 하였다.

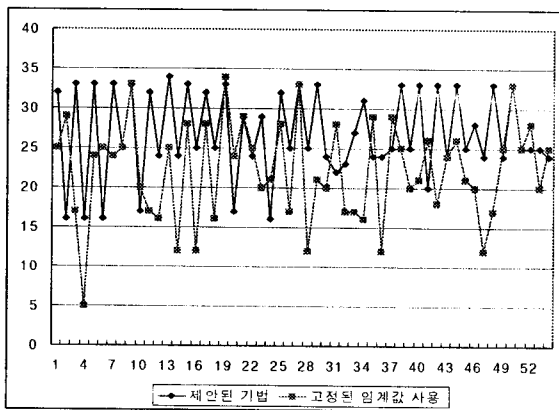


그림 4. 실험 결과

그림 4는 28000 번의 A/D 과정 중에 물체 감지가 성공한 횟수를 보여 주는 실험이다. 청색은 본 연구에서 제안된 기법을 적용한 경우의 응답특성이고 적색은 A/D변환기에 단순 임계값을 적용할 경우의 응답특성이다. 두 경우 모두 동일한 주거하에서 시뮬레이션을 수행하였으며, 시뮬레이션 결과 물체 감지 능력이 본 연구에서 제안한 기법이 우수함을 알 수 있었

다.

5. 결론

본 논문에서는 무선 센서 네트워크 하에서 이동하는 물체의 효율적인 추적을 위하여 주변 노드로부터 전송되는 측정 메시지에 기반 하여 물체의 이동을 예측한 후 센서의 임계값을 조절하여 예측한 물체의 이동을 보다 더 민감하게 감지할 수 있는 기법을 제안하였으며 시뮬레이션을 통해 제안된 기법의 유용성을 확인하였다.

본 연구는 산업자원부의 지역혁신 인력양성사업의 연구결과로 수행되었음

참 고 문 헌

- [1] Balasubramanian, S.; Elangovan, I.; Jayaweera, S.K.; Namuduri, K.R.; "Distributed and collaborative tracking for energy-constrained ad-hoc wireless sensor networks". Proceedings of WCNC 2004. Page(s):1732 - 1737 Vol.3
- [2] Yingqi Xu; Winter, J.; Wang-Chien Lee; "Prediction-based strategies for energy saving in object tracking sensor networks". Proceedings of Mobile Data Management, 2004 Page(s):346 - 357
- [3] Yang, H.; Sikdar, B.; "A protocol for tracking mobile targets using sensor networks". Proceedings of the First IEEE. 2003 Page(s):71 - 81
- [4] Xu, Y.; Winter, J.; Lee, W.-C.; "Dual prediction-based reporting for object tracking sensor networks". Proceedings of MOBIQUITOUS 2004. Page(s):154 - 163