

논문 2016-53-11-5

IoT Network에서 위치 인식을 위한 가중치 방식의 최대우도방법을 이용한 하드웨어 위치인식엔진 개발 연구

(A Hardwired Location-Aware Engine based on Weighted Maximum Likelihood Estimation for IoT Network)

김 동 순**, 박 현 문*, 황 태 호*, 원 광 호*

(Dong-Sun Kim, Hyun-moon Park, Tae-ho Hwang, and Kwang-Ho[©])

요 약

센서네트워크 센서노드의 위치정보는 기본적으로 센싱 데이터가 얻어진 위치를 알려주는 목적으로 사용되며 Context 기반 고차원 서비스를 제공하기 위한 가장 중요한 정보중 하나이다. 센서네트워크상에서 위치인식을 위해 다양한 방법들이 연구되고 제안되어 왔으며, 이러한 방법 중에 IEEE 802.15.4 센서네트워크의 물리 계층과 매체 접근 계층을 이용한 위치인식 방법에 관한 연구방법이 크게 대두되고 있다. IEEE 802.15.4 프로토콜은 장치간의 저가격, 저속의 무선 통신을 지향하기 때문에 구현에 있어서 고도화된 최적화가 중요한 요구사항이라 할 수 있다. 하지만 수신 신호의 세기를 가지고 센서 노드들의 위치를 계산하는 방법은 최적화 문제의 해를 구하기 위한 과정이기 때문에 많은 연산 량이 필요로 하게 되고, IEEE802.15.4를 지원하는 System-On-a-Chip (SoC)의 경우 8비트 마이크로 컨트롤러기반으로 설계되어 있다는 점을 고려하면, IEEE802.15.4 기반의 위치 인식 서비스를 위해서는 하드웨어에 기반을 둔 위치 인식 엔진의 필요성이 무엇보다 중요하다. 본 논문은 IEEE 802.15.4 물리계층에 기반을 둔 가중치 기반의 최대우도방법 위치인식기 하드웨어 구현에 관해 제안하고자 한다. 테스트 베드를 이용한 필드테스트 결과 제안하는 하드웨어 기반 가중치 방식의 위치 인식방법은 정확도에서 10% 정도의 개선과 함께 내장 마이크로 컨트롤러의 연산량 및 메모리 액세스를 30% 정도 감소시켜 시스템 전원소모를 줄일 수 있는 결과를 얻을 수 있었다.

Abstract

IEEE 802.15.4 is the one of the protocols for radio communication in a personal area network. Because of low cost and low power communication for IoT communication, it requires the highest optimization level in the implementation. Recently, the studies of location aware algorithm based on IEEE802.15.4 standard has been achieved. Location estimation is performed basically in equal consideration of reference node information and blind node information. However, an error is not calculated in this algorithm despite the fact that the coordinates of the estimated location of the blind node include an error. In this paper, we enhanced a conventional maximum likelihood estimation using weighted coefficient and implement the hardwired location aware engine for small code size and low power consumption. On the field test using test-beds, the suggested hardware based location awareness method results better accuracy by 10 percents and reduces both calculation and memory access by 30 percents, which improves the systems power consumption.

Keywords : IEEE802.15.4, LR-WPAN, Location Aware, IoT Sensor Network, VLSI

* 정회원, ** 평생회원, 전자부품연구원 (Korea Electronics Technology Institute)

[©] Corresponding Author (E-mail : khwon@keti.re.kr)

※ 본 연구는 산업통상자원부(MOTIE)와 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (No. 20152010103610)

Received ; September 7, 2016 Revised ; October 10, 2016

Accepted ; October 14, 2016

I. 서 론

IoT 컴퓨팅 네트워크는 인간이 생활하는 모든 환경의 사물에 컴퓨터를 장착하고 그 사물이 지능화 및 자동화되어 상호 간의 네트워크를 형성을 통해 여러 가지 서비스에 관한 정보를 주고받을 수 있는 환경을 의미한다. 이러한 IoT 기반 센서 네트워크에서 사용되는 센서와

이를 구동시키는 센서 노드는 사람이 직접 소지하여 이를 제어할 수도 있으나, 사람의 제어를 최소화하여 주변 환경에 맞는 최적의 서비스가 제공되어야 한다. 또한, 사람이 접근하기 힘든 환경에서는 직접적인 제어가 어려울 수도 있다. 다량의 센서 노드를 곳곳에 배치하여 네트워크를 구성해야 할 경우 센서 노드의 크기는 작고, 저가이어야 하며, 저 전력으로 구동되어 배터리 교체 없이 수년 동안 사용이 가능하여야 한다는 제약이 있다. 이러한 제약을 만족하기 위해 IEEE 802.15.4 같은 센서네트워크를 위한 저속의 통신 표준으로 제안되었다^[1]. IEEE 802.15.4는 무선 개인 통신망 (WPAN)의 MAC/PHY 계층을 위해 표준으로서 장치 간의 저가격, 저속도의 IoT기반 센서기반의 네트워크 통신을 지향한다. IEEE 802.15.4 표준에서 정의한 물리계층은 약 10~30미터 거리 내, 대략 250 kbps의 전송 속도를 목표하며, PHY과 MAC을 이용한 위치인지 방법에 관한 연구 및 다양한 구현이 이루어지고 있다^[1~6]. 센서네트워크 노드의 위치정보는 기본적으로 측정데이터가 얻어진 위치를 알려주는 목적으로 사용되며 또한, Context 기반 다양한 서비스를 제공하기 위한 중요한 정보 중 하나이다. 위치정보를 얻으려는 방법으로 GPS를 이용하거나 초음파, 자기센서 또는 가속도센서 등과 같은 외부 센서를 이용한 위치인식을 이용하기도 한다^[4~7]. 그 밖에도 Radio 기반에서는 Ultra wide band (UWB)에서 Time Of Arrival (TOA), Time Difference Of Arrival (TDOA) 등을 이용한 방식들이 연구되었다. 이러한 방식들에 비해 Received signal strength indication (RSSI) 기반 위치인식 방식은 무선 채널 환경에 민감하며 환경변수가 크게 영향을 줌으로써 정확한 위치인식을 요구하는 응용분야에 적용하기에는 어려움이 있다. 하지만 RSSI 기반에 위치인식 기술^[2~3, 6]은 기존 RF 모델만을 이용하여 저렴한 비용으로 적용가능하고 정확한 동기화가 요구되지 않는 점 등 제약조건이 적어 위치인식의 성능이 향상됨에 따라 그 응용분야가 확장되고 있다. 최근 ATMEL이나 IT(Texas Instruments), NXP를 볼 때 IEEE 802.15.4 표준을 지원하는 단일 칩(SoC)으로 출해되고 있으며, 이에 따라 위치추적을 위한 PHY과 MAC 최적화가 중요한 항목이라 할 수 있다. 이러한 구현에서 소프트웨어를 이용해 반복연산을 통한 최적화 알고리즘인 위치 인식은 임베디드 프로세서의 연산 처리과 시간의 증가로 인해 센서노드로써 역할을 못하는 문제점을 갖는다^[2~3, 16]. 따라서 위치인식알고리즘의 일부 기능은 구현 복잡도 대비 성능향상 고려하여 하드웨어로

구현하는 것이 프로세싱자원 측면이나 노드의 배터리 자원에 측면에 유리하다.

본 논문은 IEEE 802.15.4의 물리계층을 따르는 시스템에서 RSSI의 변화에 따라 센서 노드의 위치를 예측하는데 발생하는 오류를 최소화하기 위해 blind 노드의 위치인식에 미치는 영향에 대한 가중치를 적용하는 방법을 제안하고, 센서 네트워크의 디바이스에 사용하기 적합한 하드웨어기반의 위치인식 엔진에 관한 구현에 관해 논하고자 한다. 이를 위해 II 장에서는 제안하는 최우도기반의 위치인식 알고리즘을 설명하고, III 장에서는 제안된 하드웨어기반의 위치인식 엔진의 구조에 설명하고자 한다. IV장에서는 구현한 위치인식 엔진을 이용한 실험 결과 및 성능 분석에 관해 기술한다.

II. 가중치기반 최우도 방식의 위치인식 방법

상용화된 IEEE 802.15.4기반의 위치인식 하드웨어의 경우 주변 센서들과 자신사이의 RSSI와 TOA 정보를 이용한 위치인식을 위한 방안으로 MLE (Maximum-likelihood estimator)를 이용하였다. 이러한 Relative Location Estimation 방법은 수식(1)과 같은 MLE를 사용하여 Blind 노드의 위치에 대한 추정 값을 계산하게 된다^[3, 6, 16].

$$\hat{\theta}_R = \arg \min \sum_{i=1}^{m+n} \sum_{\substack{j \in H(i) \\ j < i}} \left(\ln \frac{\tilde{d}_{i,j}^2 / C^2}{d_{i,j}^2} \right)^2 \quad (1)$$

$\hat{\theta}_R$: Blind 노드의 좌표 추정치

$d_{i,j}$: 유클리디언 거리

$\tilde{d}_{i,j}^2$: 측정된 RSS에 의한 계산거리 (Path-Loss 모델)

m : Reference 노드 개수

n : Blind 노드 개수

H(i) : 노드 i와 링크를 가지는 노드들

여기서 MLE에서 사용되는 C는 다음과 같이 정의할 수 있다.

$$C = \exp \left[\frac{1}{2} \left(\frac{\ln 10}{10} \frac{\sigma_{dB}}{n} \right)^2 \right] \quad (2)$$

위치인식 과정은 상기 수식 (1)에 기술된 ML 함수를 최소화하는 추정값인 $\hat{\theta}_R$ 를 Blind 노드들의 위치에 대한 추정치이며, 기본적으로 Reference 노드와 Blind 노

드의 정보를 동등하게 고려하여 위치 추정에 사용한다. 하지만, 이러한 추정 방법은 Blind 노드의 위치가 추정된 좌표가 오차를 포함하게 되면 상대적으로 추정 위치의 결과도 오차가 발생하게 된다. 따라서 이러한 오차를 줄이기 위해 레퍼런스 노드 및 Blind 노드의 초깃값을 보정을 하는 추가 작업이 발생한다. 이러한 보정 작업 역시 응용 분야 또는 주위 환경에 따라 상당히 많은 제약점을 갖는다. 이러한 문제점을 개선하여 상황에 따른 보정 연산이 자유로운 가중치 기반에 하드웨어 위치 인식 알고리즘을 제안한다.

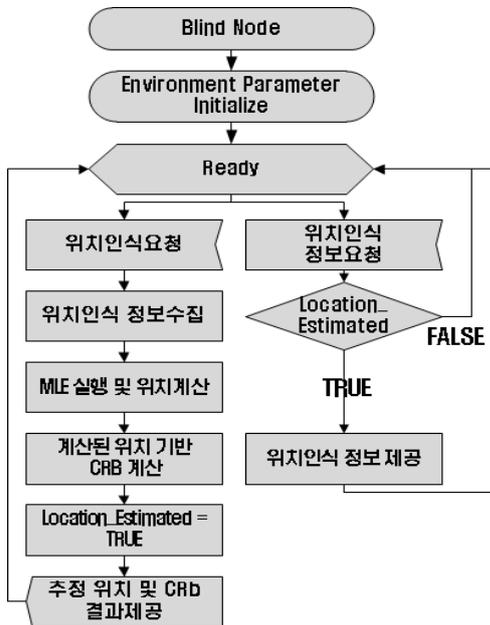


그림 1. Blind 노드를 위한 위치 추정
Fig. 1. Position estimate for the Blind node.

본 논문에서 제안된 가중치 방법은 Blind 노드의 위치를 추정하는 방법으로 최대우도 함수에서 이용되는 각 Reference 노드 및 주변 Blind 노드의 영향도를 크래머-라오 하한(Cramér-Rao lower bound, CRb)을 이용해 분산의 하한값에 따라 가중치를 두어 Blind 노드의 위치 추정에 대한 오차를 개선하였다. 이는 그림 1 과 같이 Blind 노드들의 위치 추정이 완료되었을 때 다른 노드의 위치추정에 사용되는 Collaborative 위치인식에서 기존 추정치의 Error가 새로운 노드의 위치인식에 미치는 영향에 대한 가중치를 적용하게 된다.

1. Path-Loss 모델

RSSI를 기반으로 하는 위치인식의 경우 무선 채널상의 신호 감소를 통해 대략적인 위치 값을 계산하게 되

며, 이는 추정을 위한 초기 조건으로 사용된다. 일반적으로 무선 채널상의 Path-Loss는 다음과 같이 계산할 수 있다^[4, 10].

$$P_{i,j} = \overline{p_o} - 10n \log_{10} \left(\frac{d_{i,j}}{d_0} \right) + X_\sigma \tag{3}$$

$P_{i,j}$ 는 노드 j 에 의해서 보내진 RF 신호가 노드 i 에서 받아질 때 dBm 단위의 신호세기를 나타낸다. $\overline{P_o}$ 는 d_0 에서 수신되는 평균 수신강도를 나타낸다. n은 Path-Loss exponent를 나타내고 X_σ 는 Shadowing 효과를 반영한 σ 의 표준편차를 포함한 normal random variable이다. $d_{i,j}$ 는 노드 i에서 j까지의 유클리디언 거리를 나타내며 d_0 는 $\overline{P_o}$ 를 측정할 때 사용한 기준 거리이다. 여기서는 d_0 로 1 미터를 사용하였다.

2. 최대우도추정을 위한 ML 함수

Blind 노드 θ 와 레퍼런스 노드 ρ 및 네트워크상의 모든 노드 γ 는 각각 다음과 같이 정의 된다.

$$\begin{aligned} \theta &= [\theta_1, \dots, \theta_n], i \text{ for } 1..n, \theta_i = \{x_i, y_i\} \\ \rho &= [\rho_{n+1}, \dots, \rho_{n+m}], i \text{ for } n+1..n+m, \rho_i = \{x_i, y_i\} \\ \gamma &= \{x | x \in \theta \cup \rho\}, i \text{ for } 1..n+m, \gamma_i = \{x_i, y_i\} \end{aligned} \tag{4}$$

여기서, Blind 노드 중에서 위치가 계산된 모든 노드를 각각 ϕ 로 나타내고, θ_i 노드의 이웃 노드 중에서 reference로 적용 이용 가능한 노드들의 인덱스를 $N(i)$ 로 나타내면, 가중치 기반의 위치인식 알고리즘에 사용되는 최대우도함수는 수식(5)와 같이 정의 된다.

$$\hat{\theta}_i = \operatorname{argmin}_{j \in N(i)} \left(\ln \left(\frac{\tilde{d}_{i,j}^2}{(1+e_j)C^2 d_{i,j}^2} + \frac{e_j}{1+e_j} \right) \right)^2 \tag{5}$$

수식 (5)에서 사용되는 e_j 는 MLE 계산시 각 노드의 신호에 대한 가중치 값으로 작용하게 된다. 즉 e_j 가 0 일 경우 최대의 영향을 가지게 되며 0보다 큰 값을 가질 경우 ML 함수의 영향을 줄이는 역할을 하게 된다. 개선된 가중치 기반의 MLE는 에러 범위(e_j)가 커지면 0일 때와 비교하여 실측거리($\tilde{d}_{i,j}$)와 추정거리($d_{i,j}$)값의 차로 인하여 발생하는 ML값의 영향이 그림 2의 시뮬레이션 결과에서 알 수 있듯이 줄어들게 된다. 아래 수식과 같이 계산되는 e_j 값은 레퍼런스 노드에 대해 0이

적용되며 Blind 노드의 경우에는 수식 (6)과 같이 크래머-라오 하한값에 가중치(w)를 적용한 값으로 표현되고, 가중치 값인 w 는 실험적으로 최소의 평균 위치인식 오류가 포함된 값으로 결정된다.

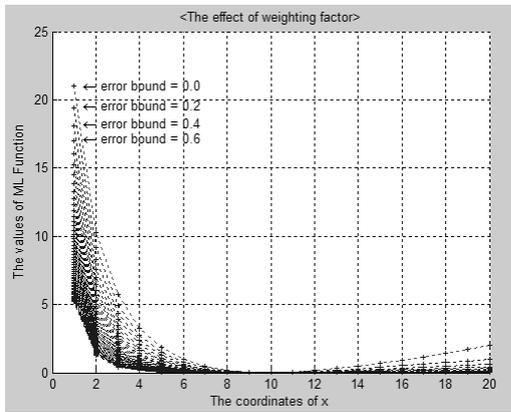


그림 2. 개선된 MLE에서 여러 바운드(e_j)의 가중치에 따른 변화

Fig. 2. Effect trends of error bound(e_j) weighting factor under the improved MLE.

$$e_i^2 = \begin{cases} 0 & \gamma_i \in \rho \\ \frac{1}{b} \frac{\sum_{j \in N(i)} d_{i,j}^{-2}}{\sum_{j \in N(i)} \sum_{k \in N(i)} \left(\frac{d_{i,j} d_{j,k}}{d_{i,j}^2 d_{i,k}^2} \right)^2} w, & \gamma_i \in \theta \end{cases} \quad (6)$$

수식 (6)에서 사용되는 계수 b 와 $d_{i \perp j, k}$ 는 다음과 같이 정의할 수 있다.

$$b = \left(\frac{10n}{\sigma_{dB} \ln 10} \right)$$

$$d_{j \perp j, k} = \frac{2 \sqrt{s(s-d_{ij})(s-d_{ik})(s-d_{jk})}}{d_{jk}} \quad (7)$$

$$s = \frac{d_{ij} + d_{ik} + d_{jk}}{2}$$

3. 최대우도추정치 계산

제시된 ML 함수를 최소화하기 위해 추정치 $\hat{\theta}_i$ 를 계산하려는 방법은 크게 두 가지 방법을 이용하게 된다. 첫 번째 방법은 ML 함수의 값을 탐색 공간 내에서 반복적 계산을 통하여 MLE를 최소화하는 $\hat{\theta}_i$ 에 대한 추정치 찾는 방법이다. 이 방법은 구현이 간단한 대신에 해를 찾는 데 많은 반복계산으로 인하여 계산 시간이 길어지며 계산 시에 적용되는 탐색 밀도에 따라서 정확도

가 떨어지는 단점이 있다. 따라서 제안한 하드웨어 기반 위치인식기는 반복적 분석방법을 통해 초기 값을 추정하려는 방안으로 사용되었다. 둘째는 NCGM (Nonlinear Conjugate Gradient Method)을 이용한 방법으로 해 탐색에 들어가는 반복계산을 최소화함으로써 계산시간을 최소화할 수 있으며 근사 최적해를 찾음으로써 정확한 위치인식 결과를 계산할 수 있는 장점이 있다. 하지만, 구현이 복잡하다는 단점이 있으며 초기 값이 잘못 선정되었을 때는 Local Optimal에 수렴하는 문제점이 있다. 이런 문제를 극복하기 위해 본 논문에서는 반복적 해 탐색을 이용한 방법과 NCGM을 이용한 방법을 조합하였고, 연산의 효율성을 하드웨어 로직을 구현하여 추론 결과를 극대화 하였다.

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)}.$$

$$d_{(0)} = r_{(0)} = -f'(x_{(0)})$$

$$d_{(i+1)} = x_{(i+1)} + \beta_{(i+1)} d_{(i)}$$

$$\beta_{(i+1)} = \max \left\{ \frac{r_{(i+1)}^T (r_{(i+1)} - r_{(i)})}{r_{(i)}^T r_{(i)}}, 0 \right\} \quad (8)$$

$$r_{(i+1)} = -f'(x_{(i+1)})$$

$$\alpha_{(i)} : \text{minimizes } f(x_{(i)} + \alpha_{(i)} d_{(i)})$$

NCGM (Nonlinear Conjugate Gradient Method)을 이용한 MLE의 해를 찾기 위해 사용된 알고리즘은 빠른 수렴의 특징을 가지는 수치최적화 알고리즘인 Polak-Ribiere를 수식(8)에 사용하였으며, 반복적인 처리를 통해 점차 초기 값의 분포를 수렴하는 전처리 과정을 거칠 수 있다.

$$\alpha = \begin{cases} \log_2(1 + \eta), & \eta \geq 0 \\ -\log_2(1 - \eta), & \eta < 0 \end{cases}$$

$$\eta = -\epsilon \frac{[f'(x)]^T d}{[f'(x + \epsilon d)]^T d [f'(x)]^T d} \quad (9)$$

$\epsilon_{[0]}$: Choose an arbitrary

$$\epsilon_{[i+1]} = -\alpha_{[i]}$$

비선형 방정식에 초기 값 분포 $\alpha_{(i)}$ 에 대해 빠른 값 계산을 위해 Line Search 알고리즘은 제안된 ML 함수에 적합하도록 수식(9)에 Secant Method를 사용하였다.

위에서 제안된 알고리즘에 사용되는 도함수 $f'(x, y)$ 는 최대우도 함수로부터 유도될 수 있으며 도함수의 x 와 y 에 관한 편미분은 수식(10)과 같이 정리하였다.

$$f(xy)' = \begin{bmatrix} \frac{\partial f(xy)}{\partial x} \\ \frac{\partial f(xy)}{\partial y} \end{bmatrix}$$

$$k = \frac{\widetilde{d}_1^2}{(1 + e_i)c^2 d_i^2} + \frac{e_i}{1 + e_i}$$

$$m_x = \frac{\widetilde{d}_1^2(x - x_i)}{d_i^2(\widetilde{d}_1^2 + e_i c^2 d_i^2)}$$

$$m_y = \frac{\widetilde{d}_1^2(y - y_i)}{d_i^2(\widetilde{d}_1^2 + e_i c^2 d_i^2)}$$

$$\frac{\partial f(xy)}{\partial x} = -4 \sum_{i=1}^n \ln k m_x$$

$$\frac{\partial f(xy)}{\partial y} = -4 \sum_{i=1}^n \ln k m_y \tag{10}$$

탐색 공간 내에서 반복적 계산을 통하여 MLE를 최소화하는 방법은 구현에서 상당히 간단한 구조를 가지나 탐색공간의 크기 및 탐색 단위에 따라서 계산량에 상당한 영향을 미치게 된다. 그림 3는 반복적 탐색 방법에 대한 간략한 흐름을 보여주고 있다.

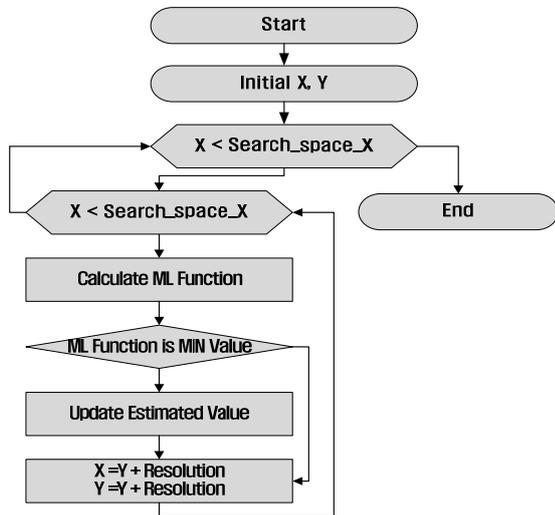


그림 3. 반복적 탐색을 통한 해의 계산 절차
Fig. 3. Value calculation procedure using recursive searching.

4. 가중치 기반 MLE 방법의 위치인식

Collaborative Location Aware 알고리즘은 Reference 노드뿐만 아니라 위치가 추정된 Blind 노드의 정보를 이용하여 위치인식의 정확도를 개선한다. Blind 노드는 자신의 위치를 추정하기 위하여 이웃노드들 중에 위치 추정이 된 Reference와 Blind 노드들을 이용하게 된다.

이러한 과정은 노드들의 위치 추정 주기에 따라서 계속 반복함으로써 이웃 노드의 업데이트 된 정보를 다시 이용하여 위치인식 결과를 개선한다. 그림 3은 제안된 Collaborative Location Aware 방법을 통한 노드들의 위치인식 결과의 정확도 개선 과정을 보여준다. 제안된 Collaborative Location Aware 알고리즘에서 교환되는 정보는 위치인식에 사용되는 노드의 측정된 RSSI 정보, 가중치를 적용한 크래머-라오 하한값, 노드의 좌표 또는 추정된 좌표 정보로 구성된다. 그림 4에서 검은 점은 Reference 노드를 흰색 점은 Blind 노드를 표시하고, 실선은 위치인식에 사용되는 노드를 나타내고 점선은 이웃 노드나 위치인식에 사용되지 않는 노드를 나타낸다. 그리고 각 노드의 주변의 점선원은 노드의 CRB 값을 보여주고 있다. 각 노드는 위치인식의 주기에 따라서 자신의 타이머에 따라서 위치인식 결과를 업데이트시킴으로써 새로운 위치를 계산하고 CRB를 다시 계산하게 된다.

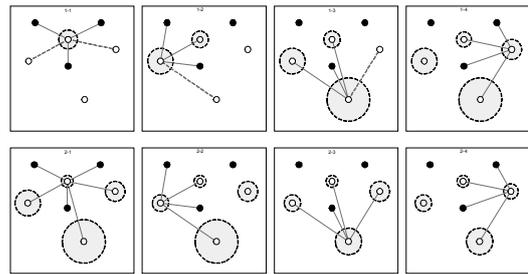


그림 4. Collaborative 위치인식 알고리즘
Fig. 4. Proposed Collaborative location-aware algorithm.

III. 가중치기반 최대우도 방식의 하드웨어 위치인식기

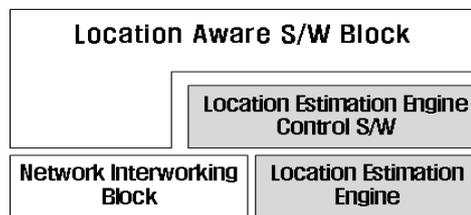


그림 5. 하드웨어기반 위치인식 엔진 구조
Fig. 5. Hardware-based location-aware engine structure.

하드웨어 기반 위치인식 기술의 전체 구조는 그림 5와 같이 크게 특정 단위 블록에서의 위치 추정을 수행하는 Location Estimation Engine 및 이를 제어하는 S/W 블록과 위치인식을 위해서 무선 네트워크와 연동하는 부분 그리고 Global Location Aware를 수행하는

소프트웨어 부분이 존재한다.

그림 5에서 Location Estimation Engine은 Blind 노드의 위치인식에 필요한 MLE를 계산하는 부분으로 ML Function의 최소값을 찾는 H/W이고, Location Estimation Engine Control S/W는 Location Estimation Engine과의 상호작용을 통해서 Blind 노드의 위치를 계산하는 S/W이다. Network Inter-working Block은 센서네트워크 또는 위치인식에 Zigbee 네트워크 등을 이용하여 위치인식에 필요한 메시지들을 주고받는 데 사용된다. Location Aware S/W 은 Location Estimation Engine을 이용하여 네트워크 안에 존재하는 Blind 노드들의 위치인식 서비스를 제공하는 블록으로 노드간의 Collaborative 위치인식을 제공하며, 그림 3의 과정을 물리적인 엔진을 통해 반복 수행함으로써 주변의 노드들의 위치인식 결과를 지속해서 반영하고 위치인식 결과의 오차를 줄인다. 또한, 단위 위치 블록의 위치인식 결과를 이용해 전체적인 위치를 계산하는 블록이다.

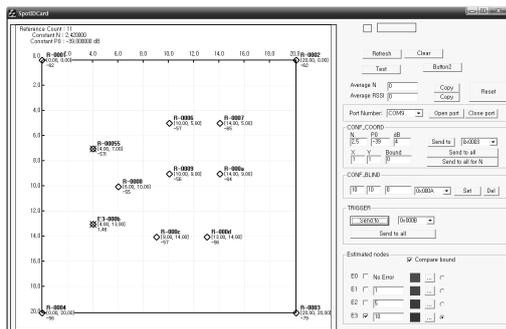


그림 6. 위치인식 하드웨어 가속기 사용한 시뮬레이션
Fig. 6. the location-simulator using location-aware accelerator hardware.

위치인식 엔진의 H/W 구현을 위해 그림 6과 같이 fixed point 시뮬레이터를 이용해 소프트웨어 블록과 연동한 동작 및 성능을 검증하였다. 시뮬레이터는 실제로 H/W 블록 및 S/W 블록의 연동을 통한 위치인식 알고리즘이 동작할 경우와 동일 조건으로 검증할 수 있도록 Fixed Point로 계산되도록 설계하였으며 하드웨어에 입력되는 환경정보를 생성하는 Data Generator와 실제 Chip 내부 Register Map과 일대일 대응되는 가상 Register Map을 통해 동작한다. 계산된 위치인식 계산 결과와 위치인식 블록의 HDL 코드의 계산 결과를 비교하는 방법으로 위치 인식 엔진 IP 설계 검증을 그림 7과 같이 진행하였다.

하드웨어 위치인식엔진은 verilog-HDL을 이용해 설계하고 TSMC 0.18 μ m를 이용해 125,651 게이트 카운트

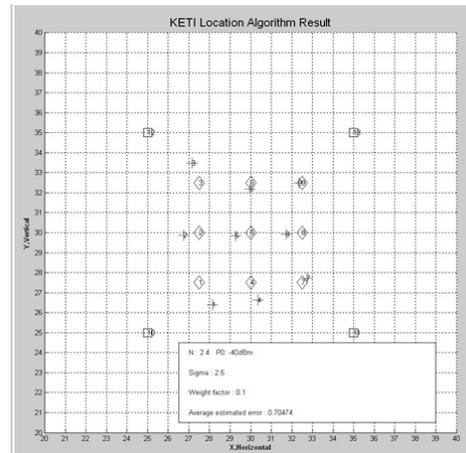


그림 7. 시뮬레이터의 시뮬레이션 결과
Fig. 7. Simulation result from the simulator.

로 구현되었다. 빠른 위치 계산을 위해 알고리즘에서 사용하는 곱셈기와 디바이더 및 로그 함수를 radix-4의 부스 알고리즘과 LUT(Look Up Table) 기반의 로그 계산 방식을 하드웨어로 설계하였다.

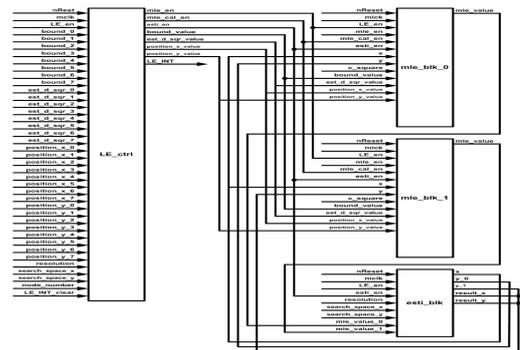


그림 8. 하드웨어 위치인식기의 블록도
Fig. 8. A block diagram of the location-aware accelerator hardware.

그림 8과 같이 하드웨어 기반 위치인식 엔진은 크게 MLE_blk과 esti_blk으로 나누어지며, MLE_blk은 초기 x와 y의 좌표값을 입력받고 각 노드에 위치한 좌표값과 각 노드간의 RSSI과의 MLE함수를 이용하여 각 노드에 MLE값을 출력한다. Esti_blk에서는 입력받은 위치정보와 각 노드의 MLE 값, 그리고 설정된 x와 y의 측정 해상도를 입력받아 최소 MLE값을 갖는 노드의 위치 정보를 참고하여 인식하려는 노드의 위치를 측정한다. 그림 9는 이와 같은 위치 인식을 실험한 결과를 나타내며, MLE블록에서 x와 y에 대한 정보와 RSSI값인 est_d_sqr_value를 입력받아 MLE값을 구한 후 estimation에서 최소값인 148,719과 x와 y의 위치 정보 7과 2를 찾는 것을 확인할 수 있다. 표 1은 24MH동작

의 128KByte Flash 메모리를 가지는 8051상에서 순수 소프트웨어 기반으로 위치인식 수행과 하드웨어기반의 위치인식 수행의 연산 시간을 비교하였다. 결과적으로 연산속도에서 제안 구조가 약 25배의 빠른 수행 결과를 보인다.

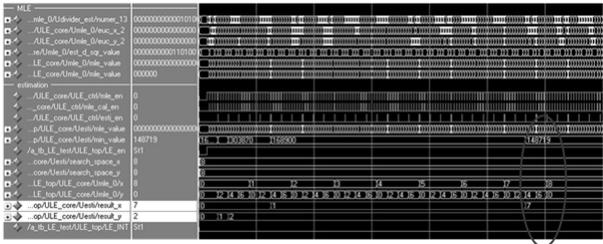


그림 9. 하드웨어 위치인식기의 검증 결과
 Fig. 9. A simulation wave of the location-aware accelerator hardware.

표 1. 연산시간비교 (sec)
 Table1. Comparison operation of the software and VLSI.

비교 항목	소프트웨어	하드웨어 (VLSI)	차이
연산 속도	12.5s	50ms	24(12.9%)
추가 로직	0	125,651	166(78.7%)

IV. 성능 평가

그림 10과 같이 개발된 위치인식 Chip을 이용하여 실제 환경에서 시뮬레이션과 동일한 20×20m의 Field Test를 진행하였으며, 그림 11과 그림 12는 상용칩^[2]과 제안된 위치인식 엔진을 사용한 시스템의 Field Test 환경을 보여 주고 있다. 그림에서 Weighted MLE를 사용할 경우와 Patwari의 Original 알고리즘을 사용했을 때 Field Test 결과를 비교하였을 때 그림 6, 7의 시뮬레이션 결과와 같이 10% 정도의 성능 개선을 보인다.



그림 10. 필드테스트 화면
 Fig. 10. Testing field configuration.

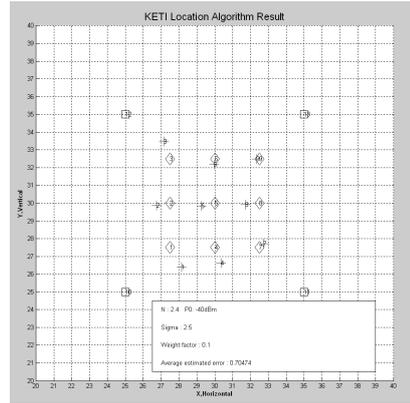


그림 11. 상용칩의 위치인식 결과 화면
 Fig. 11. Location-aware result of a commercial chip.

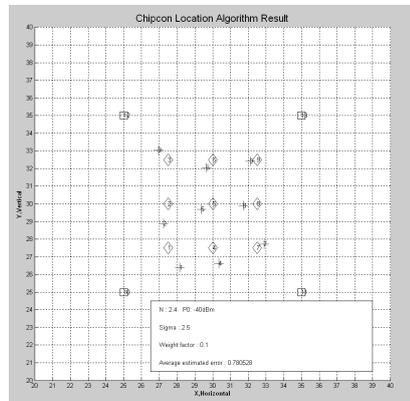
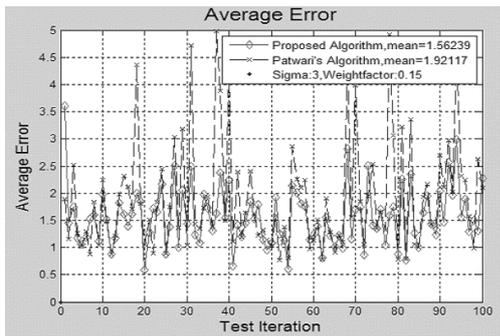


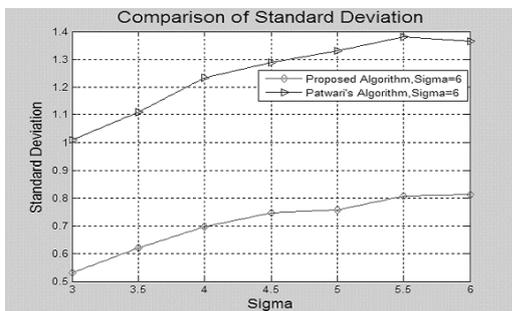
그림 12. 하드웨어 지향 위치인식엔진의 결과 화면
 Fig. 12. Location-aware result of a the proposed hardware (VLSI) chip.

V. 결론

IoT 기반의 컴퓨팅 기술은 산재한 산업 및 가정용의 다양한 스마트 기기들에 무선노드 기능을 제공하여 임의의 무선 네트워크, 정보통신 및 서비스를 가능하게 하는 IoT 기술의 가장 핵심적인 위치를 차지하고 있다. 본 논문에서는 하드웨어기반의 위치인식엔진의 구조 및 검증에 관해 기술하였으며, 가중치기반의 MLE를 사용하여 무선 채널의 변화에 대한 오류를 최소화할 수 있었으며, 소프트웨어 기반 위치인식 엔진과 비교에서와 같이 25배 이상의 빠른 위치 인식이 가능 하였다. 또한, 그림 13에서 알 수 있듯이 가중치 기반의 위치 인식 방법은 변화하는 환경에서의 표준 오차를 2배 이상 줄이는 효과를 얻을 수 있어, 더욱 정확한 위치 인식 서비스가 가능할 것으로 예상된다.



(a) 가중치 적용 MLE 평균오차
(a) the average error result of the Weighted MLE



(b) 가중치 적용 MLE 오차의 표준 편차
(b) the standard deviation result of the Weighted MLE.

그림 13. 평균오차 및 표준오차의 실험 결과
Fig. 13. Location-aware comparison result of the standard error.

REFERENCES

- [1] IEEE Std 802.15.4-2006, 'IEEE Standard for Information technology. Telecommunications and information exchange between systems Local metropolitan area networks Specific requirements, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)', 2006.
- [2] O. Hernandez, V. Jain, S. Chakravarty and P. Bhargava, "Position location monitoring using IEEE 802.15.4/ZigBee technology", in press
- [3] 김기형, 정원도, 박준성, 서현곤, 박승민, 김홍남, "IEEE 802.15.4 기반의 유비쿼터스센서네트워크 기술", 전자공학회지, 제 31권, 제 12호, 74-84쪽, 2004년 12월.
- [4] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives," in Proc. Des. Autom. Test Eur., pp.196-201, Mar. 2005.
- [5] 전호인, "IEEE 802.15.4 WPAN 기술", 전자공학회지, 제32권, 제 4호, 87-104쪽, 2005년 4월.
- [6] L. Li, W. Yang, M.Z Bhuiyan and W. Guojun, "Unsupervised learning of indoor localization based on received signal strength," Wireless Communications and Mobile Computing, pp. 2225-2237, May. 2016.
- [7] C. Medina, J.C Segura and A. Torea, "Accurate time synchronization of ultrasonic TOF measurements in IEEE 802.15.4 based wireless sensor networks", Ad Hoc Networks, Volume 11, Issue 1, pp. 442-452, Jan. 2013.
- [8] T. O. Kim, H. Kim, J. Lee, J. S. Park, and B. D. Choi, "Performance analysis of the IEEE 802.15.4 with non beacon-enabled CSMA/CA in nonsaturated condition," in Proc. Int. Conf. EUC, pp. 884 - 893, Aug. 2006.
- [9] E. Goldoni, A. Savioli, M. Risi and P. Gamba "Experimental analysis of RSSI-based indoor localization with IEEE 802.15. 4." In Wireless Conference 2010 European. IEEE, pp. 71-77, April. 2010.
- [10] Lo Bello, L.; Toscano, E., "Coexistence Issues of Multiple Co-Located IEEE 802.15.4/ZigBee Networks Running on Adjacent Radio Channels in Industrial Environments", Industrial Informatics, IEEE Transactions on Volume 5, Issue 2, Page (s).157-167, May. 2009.
- [11] S. Shin, et al, "Packet error rate analysis of IEEE IEEE 802.15.4 under between IEEE 802.15.4 and IEEE 802.11b interference," In International Conference on Wired/Wireless Internet Communications, pp. 279-288, 2005.
- [12] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, R. Han, "MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms", ACM/Kluwer Mobile Networks & Applications (MONET). Special Issue on Wireless Sensor Networks Vol. 10. No. 4, pp. 563-579, 2005.
- [13] A. Mainwaring, J.Polastre, R. Szewczyk, D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", First ACM Workshop on Wireless Sensor Networks and applications, pp. 88-97, 2002.
- [14] J. J. Lee, M. A. Jung, S. R. Lee and I-H. Son, "KNN/ANN Hybrid Location Determination Algorithm for Indoor Location Base Service", Journal of The Institute of Electronics and Information Engineers, Vol.48, No.2, pp. 109-115, Mar 2011.

[15] D. Shi, C. Tongwen and S. Ling. "Event - triggered maximum likelihood state estimation." Automatica, Vol.50, No.1 pp. 247-254, 2014.

[16] A. Coluccia, and R. Fabio, "A software - defined radio tool for experimenting with RSS measurements in IEEE 802.15. 4: implementation and applications." International Journal of Sensor Networks, Vol.14, No.3, pp. 144-154, 2013.

저 자 소 개



김 동 순(평생회원)
 1997년 인하대학교 전자재료공학 석사졸업.
 2010년 인사대학교 전자재료공학 미디어시스템 박사졸업.

1999년~현재 전자부품연구원 융합시스템연구본부 SoC플랫폼센터 센터장.
 <주관심분야: 임베디드 하드웨어, 멀티미디어 SoC Design>



박 현 문(정회원)
 2006년 국민대학교 전자통신학과 석사졸업.
 2010년 국민대학교 BIT학과 박사 졸업.
 2010년~2013년 한국전자통신연구원 선임연구원.

2014년~현재 전자부품연구원 융합시스템연구본부 SoC센터 책임연구원.
 <주관심분야: IoT시스템, 임베디드시스템, 인공지능하드웨어>



황 태 호(정회원)
 1997년 한국외국어대학교 전자공학 석사졸업.
 2013년 한국외국어대학교 컴퓨터 공학 박사졸업.

2000년~현재 전자부품연구원 융합시스템연구본부 SoC센터 팀장.
 <주관심분야: RTOS, WPAN/WBAN, 임베디드 시스템 S/W>



원 광 호(정회원)-교신저자
 2004년 중앙대학교 정보통신공학 석사졸업.
 2010년 공운대학교 전자통신학 박사 졸업.
 1991년~1997년 현대전자 정보통신연구원 주임연구원.

1997년~현재 전자부품연구원 융합시스템연구본부 IoT 플랫폼센터 센터장.
 <주관심분야: WPAN System and SoC>