

추정거리의 구간을 활용한 RSSI 핑거프린팅 기반의 실내 위치인식 알고리즘

배종현*, 백호기°

RSSI Fingerprinting-Based Indoor Localization Algorithm Using an Interval of Estimated Distance

Jonghyeon Bae*, Hoki Baek°

요약

실내 무선 네트워크에서의 위치인식 문제는 RSSI(Received Signal Strength Indicator) 핑거프린팅 기법을 활용한 연구가 활발하게 이루어져 왔다. 본 논문에서는 RSSI 핑거프린팅에 기반한 두 가지 위치인식 알고리즘으로 최대우도(ML, Maximum Likelihood) 알고리즘, 유효영역 최대우도(VAML, Valid Area Maximum Likelihood) 알고리즘을 제시하고 핵심 파라미터를 제시한다. VAML 알고리즘의 주요 파라미터들이 추정위치의 정확도에 미치는 영향을 확인하기 위해, 센서노드가 RSSI를 측정하는 횟수, 참조노드의 수, RSSI 값으로 계산한 추정거리의 범위 등 주요 파라미터의 값에 차이를 주고 시뮬레이션을 수행하여 성능을 비교하였다. 마지막으로, 제시한 두 알고리즘과 wKNN(weighted KNN) 알고리즘의 추정위치 정확도와 시간 복잡도를 비교하여 VAML 알고리즘이 wKNN 알고리즘과 ML 알고리즘에 비해 좋은 성능을 보임을 검증하였다.

키워드 : 실내 위치인식, 수신신호전력, 핑거프린팅, 경로손실 모델, 시간 복잡도

Key Words : Indoor localization, RSSI, Fingerprinting, Path loss model, Time complexity

ABSTRACT

The problem of localization in indoor wireless networks has been actively studied using RSSI(Received Signal Strength Indicator) fingerprinting techniques. In this paper, we propose two localization algorithms, ML(Maximum Likelihood) algorithm and VAML(Valid Area Maximum Likelihood) algorithm, which are based on RSSI fingerprinting, and present key parameters for them. To inspect the effect of the main parameters of the VAML algorithm on the accuracy of the estimated location, the performance of the algorithm was compared by performing simulations and varying the value of key parameters such as the number of iterations the sensor node measures RSSI, the number of reference nodes, and the range of estimated distances calculated by an RSSI value. Lastly, the accuracy of the estimated locations and time complexities of the presented algorithms and wKNN(weighted KNN) algorithm were compared to verify that the VAML algorithm shows better performance compared to the wKNN algorithm and the ML algorithm.

* 본 논문(저서)은 교육부 및 한국연구재단의 4단계 BK21 사업(경북대학교 컴퓨터학부 지능융합 소프트웨어 교육연구단)으로 지원된 연구임 (4199990214394).

※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성사업의 연구결과로 수행되었음. (IITP-2024-RS-2022-00156389)

• First Author : Kyungpook National University, School of Computer Science and Engineering, bae.jh.dan@gmail.com, 학생회원

° Corresponding Author : Kyungpook National University, School of Computer Science and Engineering, neloyou@knu.ac.kr, 종신회원
 논문번호 : 202311-141-C-RU, Received November 13, 2023; Revised December 4, 2023; Accepted December 11, 2023

1. 서 론

실내 무선 네트워크 환경에서 위치기반 서비스(LBS, Location-Based Service)를 지원하기 위한 위치인식 문제는 오랜 시간 동안 주목받은 연구주제이다. 실내환경에서는 GNSS(Global Navigation Satellite System) 위성신호의 수신강도 저하 등의 제약이 있어, 네트워크의 기반 장치로부터 수신된 신호를 통해 위치와 관련된 파라미터를 측정하여 대상의 위치를 직접 추정하는 기법들이 연구되었다. 이때 사용되는 파라미터는 신호가 위치인식의 대상이 되는 노드에 도달하기까지의 전파시간을 측정한 ToA(Time of Arrival)와 측정된 ToA 간의 차이인 TDoA(Time Difference of Arrival) 등 시간측정 기반의 파라미터가 있고, 신호의 수신전력을 측정한 RSSI(Received Signal Strength Indicator)나, 참조신호가 센서노드에 도달한 입사각 AoA(Angle of Arrival) 또는 DoA(Direction of Arrival)가 활용될 수 있다.

위치인식 시스템의 하드웨어 요구사항과 제약은 활용되는 파라미터에 따라 다르다¹⁾. ToA와 TDoA는 참조노드와 센서노드 장치 간의 정밀한 시각 동기화가 필요하다. TDoA 측정 장치 간 시각 동기화 요구조건을 완화하기 위해, 알려진 위치에 고정 배치된 별도의 reference tag를 활용하여 참조노드간의 clock offset을 보상하는 실내 위치인식 시스템 구조가 제안되기도 했다²⁾. AoA를 활용하려면 배열 안테나가 필요하고, 신호가 반사되어 도달한 다중경로 성분의 입사각이 함께 측정되는 ghosting 현상을 배제해야 하는 어려움이 있다³⁾. 임베디드 장치로 구성된 실내 IoT 환경에서는 저가의 센서와 장치가 주로 활용되므로, 비용 측면에서 하드웨어 성능의 제약을 고려해야 한다. RSSI는 측정치의 정확도와 사용 가능한 거리가 상대적으로 떨어지는 단점이 있지만, RSSI 측정을 위해 별도의 장치가 필요하지 않다는 장점을 가진 파라미터이다. RSSI의 이러한 장점에 힘입어, 실내에서 BLE(Bluetooth Low Energy)를 활용한 RSSI 기반 삼변측량을 수행하기 위해, 수신기의 높이와 추정위치의 오차의 관계를 활용한 단순한 계산식으로 추정거리를 갱신하는 알고리즘이 제안되었으며⁴⁾, 무선 센서 네트워크(WSN, Wireless Sensor Network) 환경에서 RSSI와 AoA를 함께 활용하거나⁵⁾ RSSI와 ToA를 함께 활용⁶⁾ 하이브리드 형태의 파라미터에 기반한 위치인식 기법들이 활발하게 연구되었다.

실내 위치인식 연구에서 많은 관심을 받는 RSSI 핑거프린팅 방식은 센서노드가 측정한 RSSI 값을 데

이터베이스에 저장된 핑거프린트 지도(데이터셋)와 비교하고, 측정값과 가장 유사한 값을 갖는 지점을 활용하여 센서노드의 추정위치를 도출한다. 인공지능에 기반한 WiFi RSSI 핑거프린팅 기법의 기술요소를 분석하고 다양한 기법들의 성능을 비교한 연구⁷⁾는 벤치마킹이 가능한 표준 RSSI 데이터셋의 부재를 지적하였으며, 실내 위치인식에 특화된 저비용 기계학습 알고리즘의 필요성을 제시하였다. 또한, 가장 널리 활용되는 기본적인 RSSI 핑거프린팅 알고리즘인 NN(Nearest Neighbor), KNN(K-Nearest Neighbor), wKNN(weighted K-Nearest Neighbor), Bayesian 알고리즘을 대상으로 추정위치 정확도가 비교되기도 하였다⁸⁾.

EZ 알고리즘은 핑거프린팅 시스템의 초기 구축 비용(pre-deployment effort)을 해소하고자 클라우드 소싱 패러다임을 제시하였다⁹⁾. 핑거프린트 지도를 구축하기 위해 많은 위치에서 RSSI를 측정하고 각 측정치가 획득된 실제 위치를 직접 수집하는 전통적인 핑거프린팅 방식과 달리, EZ 알고리즘은 스마트폰 등 사용자 장치들이 이동하는 동안 서버의 요청에 의해 수집된 다량의 RSSI 측정치와 관측된 AP(Access Point)들의 정보를 통해 과결정(overdetermined) 연립방정식을 구성하고 최적해를 구하여 위치인식 모델을 학습시킨다. 그러나, EZ 알고리즘은 연립방정식을 구성하는 일부 데이터의 실제 측정 위치를 (창문이나 출입구 근처에서) GPS 수신을 통해 획득할 수 있어야만 위치인식 모델을 완성할 수 있다는 제약이 있으며, 중복된 정보를 포함하는 AP를 배제하여 연립방정식을 구성하는 별도의 과정을 거쳐야 한다.

RSSI와 별도의 측정치를 결합한 하이브리드 형태의 파라미터를 핑거프린팅 시스템의 입력으로 활용하는 방법이 연구되기도 했다. WiFi AP의 MAC 주소와 RSSI 측정값을 visual feature와 결합시킨 Wi-Vi 핑거프린팅 기법이 제안되었으나¹⁰⁾, 건물 내에 고르게 배치된 출구 표지판들을 랜드마크로 활용할 수 있는 환경을 전제로 분류모델을 학습하며, 위치인식시 시각적 특징(visual feature) 매칭에 별도의 컴퓨팅 자원(computational resources)을 소모해야 한다는 단점이 있다. 또 다른 연구에서는 DNN(Deep Neural Network)을 학습시켜 위치인식을 수행하였다¹¹⁾. 위치인식 모델을 학습시키기 위해, 실내 공간을 격자 형태로 나누어 각 격자점의 위치에서 모바일 장치가 다수의 AP로부터 RSSI를 측정한다. 신호 각각의 채널 상태정보(CSI, Channel State Information)들을 계산하면 측정된 RSSI와 가장 높은 상관관계를 가지는

CSI amplitude를 찾을 수 있다. 이를 RSSI 측정치와 결합하고 DAE(Deep Auto Encoder)를 학습시킴으로써 offline phase에서 데이터베이스를 구축하고 DNN을 학습시켰다. 학습이 완료된 online phase에서는 모바일 장치가 RSSI를 측정하고 CSI를 계산하여 서버에 전달하고, 학습된 DAE와 DNN을 통해 서버가 추정위치를 찾는 방식이다.

앞서 소개한 RSSI 핑거프린팅 기법들 중 일부는 위치인식 모델을 생성하기 위해 GPS로 획득한 실제 측정 위치를 포함한 데이터를 활용하거나⁹⁾, RSSI에 다른 종류의 측정치를 결합한 하이브리드 형태의 파라미터를 시스템의 입력으로 활용하므로 별도의 측정 및 계산 과정이 필요함^{10,11)}을 알 수 있다.

본 논문에서는 RSSI 외에 별도의 측정치를 활용하거나 하이브리드 형태의 측정치를 시스템 입력으로 활용하지 않으며, 데이터베이스에 저장된 RSSI 데이터의 분포와 센서노드의 RSSI 측정값만을 활용하여 추정위치를 계산하는 두 가지 핑거프린팅 알고리즘을 제시한다. 2장에서는 최대우도(ML, Maximum Likelihood) 알고리즘을 먼저 제시하고, ML 알고리즘의 시간 복잡도를 개선한 유효영역 최대우도(VAML, Valid Area Maximum Likelihood) 알고리즘을 함께 제시한다. 3장에서는 LR-WPAN(Low Rate Wireless Personal Area Networks) 환경에서 VAML 알고리즘을 적용하여 위치인식 시스템을 설계하기 위한 파라미터들을 제시하고, 시뮬레이션 결과를 통해 주요 파라미터들이 추정위치의 정확도에 미치는 영향을 확인한다. 마지막으로, 제시한 두 가지 알고리즘의 추정위치 정확도와 시간 복잡도를 wKNN 알고리즘과 비교하여 검증한다.

II. 제시하는 위치인식 알고리즘

본 장에서는 제시할 알고리즘을 뒷받침하는 시스템 구성을 설명하고, 두 가지 알고리즘을 제시한다.

2.1 시스템 구성

위치인식이 수행되는 공간은 직사각형 형태의 2차원 공간을 고려한다. 2차원 공간을 격자 형태로 나누는 행의 수와 열의 수를 정하면 일정한 간격을 갖는 격자점들의 위치는 행의 수 N_{row} 와 열의 수 N_{col} 에 의해 정의되는 격자점들의 인덱스로 표현된다. 두 정수 i, j ($0 \leq i \leq N_{row} - 1, 0 \leq j \leq N_{col} - 1$)에 대해, i 행 j 열에 위치한 격자점의 위치를 $gridpoint_{ij}$ 로 정의

한다. 그림 1은 N_{row} 와 N_{col} 을 11로 정했을 때 정의되는 격자점들을 예시로 나타낸 것이다. 이 공간에서 미터 단위의 가로 길이를 w , 세로 길이를 h 라고 하면 $gridpoint_{ij}$ 의 좌표는 다음과 같다.

$$gridpoint_{ij} = \left(\frac{h}{N_{row} - 1}i, \frac{w}{N_{col} - 1}j \right). \quad (1)$$

식 (1)의 행의 수를 $N_{row} = h + 1$ 로, 열의 수를 $N_{col} = w + 1$ 로 정하면 상하, 좌우로 인접한 격자점들이 1m 간격을 갖게 된다. 그림 1의 예시에서 h 와 w 가 각각 10이면 그림 1에서 인접한 격자점들은 상하, 좌우로 1m의 간격을 갖게 된다.

그림 1의 R_1, R_2, R_3, R_4 는 참조노드를 나타낸 것이고, id 값으로 인덱스 k 를 가지는 참조노드는 R_k 로 표현될 수 있다. k 는 $1 \leq k \leq N_{ref}$ 의 범위를 갖는 자연수이고, 여기서 N_{ref} 는 참조노드의 수를 일반화하여 표기한 것이다. 그림 1에서 $N_{ref} = 4$ 이다. 참조노드에는 전방향 송신 안테나가 탑재되어 있어, 고정된 위치에서 주기적으로 참조 신호를 전방향으로 고르게 전송한다. 각 참조 신호의 내용은 참조노드의 id와 그 참조노드가 설치된 위치의 2차원 좌표로 구성되며, 다수의 참조노드가 전송한 참조 신호들의 패킷 충돌이 일어나지 않도록 참조노드 간에는 시각 동기화가 되어있다고 가정한다. 본 논문에서는 4개 이상의 참조노드를 활용하는 시나리오를 가정하였다.

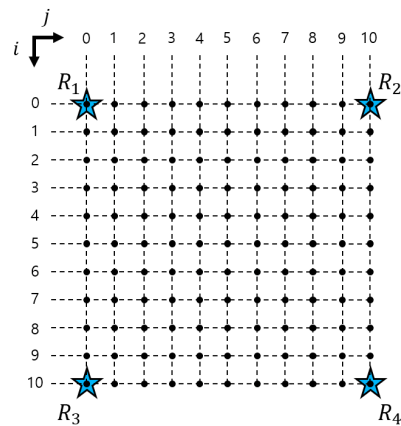


그림 1. 행, 열의 수에 의해 정의된 격자점들과 참조노드의 배치 예시 ($N_{row} = N_{col} = 11$)
 Fig. 1. An example of arrangement of reference nodes and grid points defined by the number of rows and columns ($N_{row} = N_{col} = 11$)

데이터베이스에는 정의된 모든 격자점에서 각 참조 노드별로 RSSI 측정치의 표본 평균과 표본 표준편차가 저장되어 있다. $gridpoint_{i,j}$ 에서 측정된 RSSI의 표본 평균을 $\bar{p}_k(i,j)$, 표본 표준편차를 $s_k(i,j)$ 라고 정의한다. 또한, $gridpoint_{i,j}$ 에서 수집된 RSSI의 표본은 정규분포 $N(\bar{p}_k(i,j), s_k^2(i,j))$ 를 따른다고 가정한다. $gridpoint_{i,j}$ 에서 R_k 에 대한 RSSI 측정치의 표본의 분포를 확률밀도함수로 나타내면 그림 2와 같다.

센서노드에는 RSSI를 측정할 수 있는 전방향 수신 안테나가 탑재되어 있고, 수신한 각 참조 신호로부터 참조노드의 id를 통해 다수의 참조 신호를 구분하여 RSSI 측정치를 저장할 수 있다고 가정한다. 어떤 위치에서 센서노드가 자기 위치를 찾고자 RSSI를 측정할 때, 센서노드는 측정값의 일시적인 오차를 완화할 수 있도록 고정된 위치에서 각 참조노드로부터 m 회 걸쳐 측정된 RSSI 값의 평균을 사용한다. 센서노드가 R_k 로부터 m 번 수신한 참조신호의 평균 RSSI를 p_k 라고 하면, N_{ref} 개의 참조노드로부터 각 m 개의 RSSI 표본에 대한 표본 평균을 얻을 수 있다. 즉, p_k 는 $N(\bar{p}_k(i,j), s_k^2(i,j))$ 의 확률분포를 따르는 표본 m 개의 표본 평균이므로, p_k 는 $N(\bar{p}_k(i,j), \frac{s_k^2(i,j)}{m})$ 를 따르는 확률변수라고 가정한다. 이로써 어떤 센서노드가 일정 시간 동안 측정된 값을 다음과 같이 p 로 표현할 수 있다.

$$p = [p_1 \ p_2 \ \dots \ p_{N_{ref}}]. \quad (2)$$

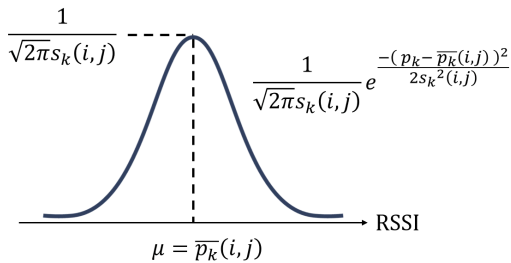


그림 2. 데이터셋의 $gridpoint_{i,j}$ 에서 측정된 R_k 로부터의 참조신호의 RSSI 표본 분포를 나타내는 확률밀도함수
Fig. 2. The probability density function that shows a distribution of RSSI samples in dataset from R_k measured at $gridpoint_{i,j}$

2.2. 위치추정 알고리즘

2.2.1 최대우도(ML) 알고리즘

본 항에서 보이하고자 하는 ML 알고리즘은 센서노드가 측정된 p 를 데이터베이스에 저장된 모든 격자점의 위치, 즉 모든 순서쌍 (i,j) 에 대한 p_k 의 우도를 구하여 그 중 최댓값을 갖는 격자점의 위치 $gridpoint_{i,j}$ 를 찾는 방식이다.

센서노드가 측정된 p_k 가 $gridpoint_{i,j}$ 에서 측정되었을 우도는 정규분포 $N(\bar{p}_k(i,j), \frac{s_k^2(i,j)}{m})$ 의 확률밀도함수의 값이다. 특정된 k 에 대한 p_k 의 단일 우도 $f(p_k|i,j)$ 는 다음과 같다.

$$f(p_k|i,j) = \frac{1}{\sqrt{\frac{2\pi}{m}} s_k(i,j)} e^{-\frac{(p_k - \bar{p}_k(i,j))^2}{2(\frac{s_k^2(i,j)}{m})}}. \quad (3)$$

p 를 구성하는 모든 확률변수 $p_1, p_2, \dots, p_{N_{ref}}$ 가 서로 독립이므로, $p = [p_1 \ p_2 \ \dots \ p_{N_{ref}}]$ 가 $gridpoint_{i,j}$ 에서 측정되었을 우도는 식 (3)의 단일 우도를 모든 k 값에 대해 각각 구하고 이들을 모두 곱하여 로그를 취한 log likelihood로 나타낼 수 있다. 이를 $L(p,i,j)$ 로 정의하면 다음과 같이 나타낼 수 있다.

$$\begin{aligned} L(p,i,j) &= \log(f(p_1, p_2, \dots, p_{N_{ref}}|i,j)) \\ &= \log\left(\prod_{k=1}^{N_{ref}} f(p_k|i,j)\right) = \sum_{k=1}^{N_{ref}} \log(f(p_k|i,j)). \end{aligned} \quad (4)$$

모든 격자점에 대해 데이터베이스에 저장된 각 참조노드로부터의 RSSI의 확률 분포를 활용하면 식 (4)에 의해 모든 순서쌍 (i,j) 에 대한 $L(p,i,j)$ 를 계산할 수 있다. $L(p,i,j)$ 가 최대가 되게 하는 순서쌍 (i,j) 를 찾으면 그 순서쌍에 해당되는 격자점의 위치 $gridpoint_{i,j}$ 를 센서노드의 추정위치로 선택할 수 있다. 따라서, 센서노드의 측정값 p 를 추정위치의 행 번호와 열 번호의 순서쌍으로 대응시키는 함수 $y(p)$ 를 다음과 같이 정의할 수 있다.

$$y(p) = \arg \max_{(i,j)} \{L(p,i,j)\}. \quad (5)$$

2.2.2 유효영역 최대우도(VAML) 알고리즘

2.2.1에서 제시한 ML 알고리즘은 모든 격자점에서 $L(p,i,j)$ 를 계산하고 최댓값을 찾기 위해 불필요하게 많은 연산이 필요하며, 시간 복잡도 측면에서도 개선의 여지가 있다. 본 항에서는 ML 알고리즘을 변형하여, 일부 격자점에서만 $L(p,i,j)$ 를 계산하고 추정위치를 도출할 수 있도록 시간 복잡도를 줄인 VAML 알고리즘을 제시한다. ML 알고리즘은 데이터 베이스에 저장된 데이터셋으로부터 $gridpoint_{ij}$ 의 RSSI 표본의 분포에 관한 정보만을 활용하고, 측정된 p_k 를 통해 R_k 로부터의 거리를 계산하지 않았다. 본 항에서 제시할 VAML 알고리즘은 R_k 로부터 수신한 참조신호에서 R_k 의 위치 정보와 p_k 를 함께 활용한다.

경로손실(PL, Path Loss) 모델은 신호의 전파거리 d 에 대한 PL값의 함수이므로, 그 역함수를 구하면 어떤 신호의 수신전력에 대한 PL값을 신호의 전파거리로 대응시킬 수 있다. 모든 참조노드의 송신전력이 p_t 로 동일하다고 가정하고, 참조신호의 수신전력을 일반화하여 p_r 이라 하면 다음과 같이 PL값에 대한 신호의 전파거리 d 를 함수 g 로 표현할 수 있다.

$$d = g(PL), \quad PL = p_t - p_r. \quad (6)$$

식 (6)은 측정된 p_r 값에 오차가 없을 때에만 등식 $d = g(PL)$ 이 성립한다. 즉, 직선거리로 d 만큼 실제로 전파된 신호의 측정된 p_r 을 통해 계산한 PL값이 경로손실 모델에 의한 d 의 함수값과 일치하는 경우에만 $d = g(PL)$ 이라고 할 수 있다. 그러나 본 알고리즘에서 활용하는 p_k 는 반사된 신호에 의한 다중경로 페이딩 및 장애물에 의한 음영효과가 포함된 RSSI 측정값이다. 무작위적인 오차가 포함된 p_k 로부터 계산된 경로손실 값을 $\widehat{PL} = p_t - p_k$ 로 표기하여 식 (6)을 고쳐 쓰면 p_k 를 통해 추정된 참조신호의 전파 거리 \hat{d} 은 다음과 같이 정의된다.

$$\hat{d} = g(\widehat{PL}), \quad \widehat{PL} = p_t - p_k. \quad (7)$$

측정된 p_k 를 통해 계산한 \hat{d} 이 참조신호의 실제 전파거리 d 와 일치할 확률은 매우 작다. p_k 는 센서노드에 의해 측정된 RSSI 표본 m 개의 평균이므로, p_k 에

존재하는 오차의 확률분포가 0을 평균으로 갖는다고 가정하면 중심극한정리에 따라 m 이 커질수록 식 (7)에 의해 계산되는 \hat{d} 이 d 에 가까워짐을 고려할 수 있다. 충분히 큰 측정 횟수 m 을 보장하여 \hat{d} 값 자체를 활용하는 방법은 센서노드가 RSSI를 측정하는데 소요하는 시간이 길어지는 단점이 있으므로, \hat{d} 값 주위의 추정거리의 구간을 통해 센서노드가 존재할 수 있는 영역만을 유효한 영역으로 고려한다.

\hat{d} 주위의 추정거리의 구간은 \hat{d}_{near} 와 \hat{d}_{far} 에 의해 정의된다. \hat{d}_{near} 와 \hat{d}_{far} 는 데이터셋으로부터 모든 i, j, k 에 대해 미리 계산된 표본 표준편차 $s_k(i,j)$ 의 평균값 σ 와, 추정거리의 구간 길이를 조절하기 위한 양의 실수 c 에 의해 다음과 같이 정의된다.

$$\hat{d}_{near} = g(\widehat{PL} - c\sigma), \quad \hat{d}_{far} = g(\widehat{PL} + c\sigma). \quad (8)$$

그림 3은 \hat{d} 과 \hat{d}_{near} , \hat{d}_{far} 의 한 예시이다. 위치인식 이 수행될 공간에서 R_k 로부터 \hat{d}_{near} 이상 \hat{d}_{far} 이하의 거리를 갖는 영역을 유효영역 A_k 로 정의한다. 그림 3에서 색칠된 영역이 A_k 를 나타낸 것이다. 여기서 k 는 $\arg \max_k \{p_k\}$ 를 선택한다. RSSI 측정치는 전파 신호의 거리가 멀어질수록 정확도가 감소하기 때문에, 가장 가까운 참조노드로부터의 수신전력인 $\max \{p_k\}$ 를 통해 구한 유효영역 A_k 가 가장 신뢰할 만하다고 할 수 있다.

A_k 를 특정하기 위해, 센서노드의 실제위치가 A_k 에 포함될 확률이 커지도록 하는 적절한 상수 c 를 정해야 한다. c 의 값이 클수록 실제 전파거리 d 가 구간 $[\hat{d}_{near}, \hat{d}_{far}]$ 에 속할 확률이 높아지지만, A_k 의 영역이 넓어지면서 A_k 에 포함되는 격자점의 수도 많아지기 때문에 연산량이 증가하는 단점이 있다. 반대로, c 의 값이 작을수록 d 가 구간 $[\hat{d}_{near}, \hat{d}_{far}]$ 에 속할 확률이 낮아지고 연산량은 감소한다.

σ 와 c 의 값에 의해, 식 (7)과 식 (8)로부터 유효영역 A_k 를 특정할 수 있다. A_k 에 포함되는 격자점들은 R_k 로부터의 거리가 \hat{d}_{near} 이상 \hat{d}_{far} 이하이다. R_k 와 $gridpoint_{ij}$ 사이의 거리를 $dist(R_k, i, j)$ 로 정의하면 모든 격자점에 대해 $dist(R_k, i, j)$ 를 계산할 수 있다 ($0 \leq i \leq N_{row} - 1, 0 \leq j \leq N_{col} - 1$). 어떤 격자점 $gridpoint_{ij}$ 에 대해, $\hat{d}_{near} \leq dist(R_k, i, j) \leq \hat{d}_{far}$ 이

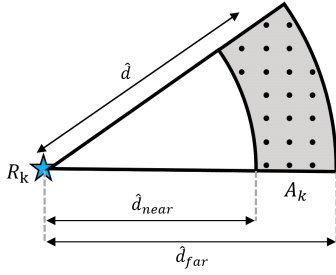


그림 3. 측정된 p_k 값과 PL모델에 의해 계산되는 \hat{d} , \hat{d}_{near} , \hat{d}_{far} 과 구간 $[\hat{d}_{near}, \hat{d}_{far}]$ 에 의해 정의된 유효영역 A_k

Fig. 3. \hat{d} , \hat{d}_{near} , \hat{d}_{far} calculated by given PL model and valid area A_k defined by the interval $[\hat{d}_{near}, \hat{d}_{far}]$ for a measured p_k value

면 $gridpoint_{i,j}$ 는 A_k 에 포함되는 격자점이다.

A_k 에 포함되는 격자점들에 대해서는 식 (4)와 같은 방식으로 $L(p, i, j)$ 를 구하고, A_k 에 포함되지 않는 격자점들의 $L(p, i, j)$ 는 0으로 정의한다. 마지막으로, 식 (5)의 함수 y 와 동일한 방식으로 $\arg \max_{(i,j)} \{L(p, i, j)\}$ 에 해당되는 $gridpoint_{i,j}$ 를 센서노드의 추정위치로 선택한다. 그림 4는 본 항에서 제시한 VAML 알고리즘의 순서도를 나타낸 것이다.

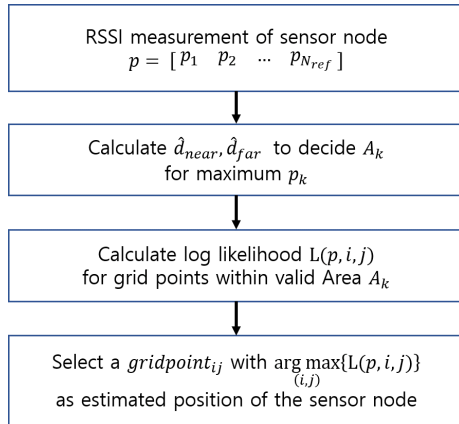


그림 4. VAML 알고리즘의 순서도
Fig. 4. Flow chart of the VAML algorithm

III. 시뮬레이션 환경 및 결과

본 장에서는 시뮬레이션 환경을 소개하고, 본 논문에서 제시하는 알고리즘을 적용한 실내 위치인식 시스템의 설계에 고려되어야 하는 파라미터를 설명한

후에 시뮬레이션 결과들을 제시한다.

시뮬레이션 환경은 IEEE 802.15.4a의 표준^[12]을 따르는 LR-WPAN을 가정하였으며, PL모델은 다음과 같다.

$$PL = pl(d) = \begin{cases} 40.2 + 20 \log_{10}(d) & , d \leq 8[m] \\ 58.5 + 33 \log_{10}(\frac{d}{8}) & , d > 8[m] \end{cases} \quad (9)$$

식 (6)과 식 (7)의 함수 g 는 함수 $pl(d)$ 의 역함수이므로, 다음과 같이 구할 수 있다.

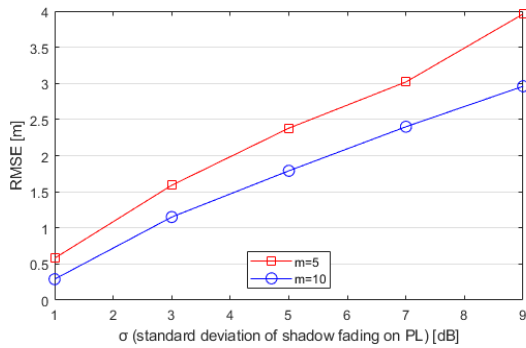
$$d = g(PL) = \begin{cases} 10^{\frac{(PL-40.2)}{20}} & , PL \leq 58.5 [dB] \\ 8 \times 10^{\frac{(PL-58.5)}{33}} & , PL > 58.5 [dB] \end{cases} \quad (10)$$

표 1은 본 논문에서 제시한 VAML 알고리즘의 파라미터를 정리한 표이다. 후술할 모든 시뮬레이션 결과는 데이터베이스의 각 격자점별 표본 데이터 수 n 을 100으로 설정한 결과이며, 음영효과에 의한 PL값의 표준편차 σ 는 데이터셋과 센서노드에 동일하게 적용하였다.

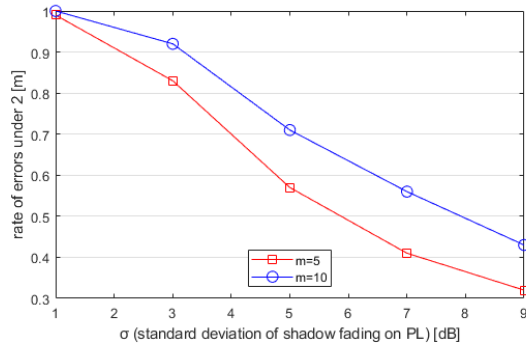
그림 5는 VAML 알고리즘을 수행할 때, m 이 추정 위치의 정확도에 미치는 영향을 확인하기 위한 그래프이다. 공통적으로 $w = h = 20$ 인 정사각형 공간에서 $N_{row} = N_{col} = 21$ 로 설정하여 인접한 격자점들이

표 1. 제시하는 알고리즘의 파라미터들과 그 설명
Table 1. Parameters of the presented algorithm and their descriptions

Parameter	Description
w	Width of the 2-D space [m]
h	Height of the 2-D space [m]
N_{row}	Number of rows
N_{col}	Number of columns
N_{ref}	Number of reference nodes
n	Number of RSSI samples at each grid point(in the dataset from database)
m	Number of iterations for which RSSI measurement(from each reference node) is performed by a sensor node
σ	Average standard deviation of error caused by shadow fading on PL values [dB]
c	Positive real number to determine the length of interval $[d_{near}, d_{far}]$ in (8)



(a) RMSE by two different values of a parameter m ($m = 5, m = 10$)



(b) Rate of errors within 2 m in estimated positions by two different values of a parameter m ($m = 5, m = 10$)

그림 5. 센서노드의 측정 횟수에 따른 추정위치의 정확도 비교

Fig. 5. Comparison of estimated position accuracy according to the number of measurement iterations

1m의 간격을 가지며, $N_{ref} = 4$ 이고 참조노드는 그림 1과 같이 실내 공간의 각 꼭지점에 배치하였다. c 의 값 또한 공통적으로 1.5로 설정하였다.

그림 5(a)와 그림 5(b)의 가로축은 공통적으로 음영 효과의 정도를 표현하는 σ 의 값으로, σ 가 1, 3, 5, 7, 9 dB인 다섯 가지 환경에서 시물레이션을 수행하였다. 그림 5(a)의 세로축은 존재하는 모든 격자점에서 알고리즘을 수행하여, 실제 위치와 추정된 위치 사이의 오차에 대해 계산한 RMSE(Root Mean Squared Error)이다. 즉, 시물레이션 결과의 RMSE는 센서노드가 모든 격자점에서 한 번씩 알고리즘을 수행하도록 반복하여 산출한다. RMSE의 계산식을 명확히 표현하기 위해 식 (2)의 p 를 일반화하여, i 행 j 열의 격자점에서 알고리즘을 수행할 때의 p 를 $p(i, j)$ 로 바꿔 표기하고, 식 (5)의 함수 y 와 관련하여 $y(p(i, j))$ 를 통해 구한 격자점을 실제 좌표로 대응시키는 함수를 $Y(i, j)$ 로 정의하면 본 장에서 보이고자 하는 RMSE의 계산식은 다음과 같다.

$$RMSE = \sqrt{\frac{\sum_i \sum_j (Y(i, j) - \text{gridpoint}_{ij})^2}{N_{row} N_{col}}} \quad (11)$$

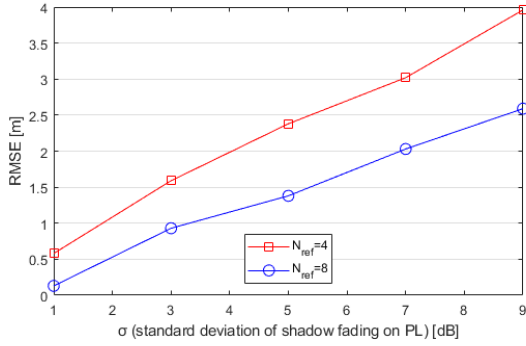
그림 5(b)의 세로축은 위치인식이 수행되는 모든 격자점 중에서 추정위치의 오차거리가 (실내 공간의 한 변의 길이 $l = w = h$ 의 1/10인) 2.0m 이하인 격자점들의 비율을 나타낸 것으로, 그 계산식은 다음과 같다.

$$\frac{\left| \{(i, j) | \sqrt{(Y(i, j) - \text{gridpoint}_{ij})^2} \leq \frac{l}{10}\} \right|}{N_{row} N_{col}} \quad (12)$$

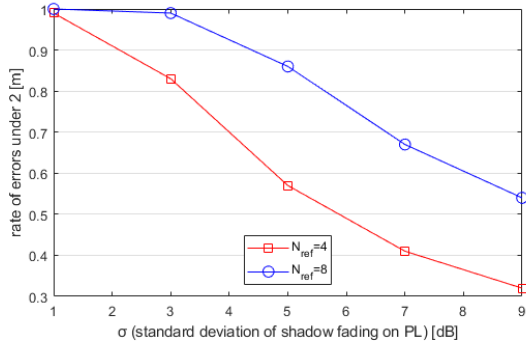
그림 5(a)에서 모든 σ 값에 대해 $m = 10$ 으로 설정한 결과가 $m = 5$ 로 설정한 결과에 비해 추정위치의 RMSE가 작았으며, 그림 5(b)에서 모든 σ 값에 대해 $m = 10$ 으로 설정한 결과가 $m = 5$ 로 설정한 결과에 비해 2m 이하의 오차가 발생한 격자점의 비율이 높았다. 이것은 $p_k \sim N(\bar{p}_k(i, j), \frac{s_k^2(i, j)}{m})$ 이므로 m 이 클수록 p_k 가 $\bar{p}_k(i, j)$ 에 가까워질 확률이 높기 때문임을 알 수 있다.

그림 6은 참조노드가 많을수록 VAML 알고리즘의 추정위치가 정확해지는지 확인하기 위한 그래프이다. N_{ref} 에만 4와 8로 차이를 두고, $w = h = 20$, $N_{row} = N_{col} = 21$, $c = 1.5$, $m = 5$ 등 N_{ref} 를 제외한 나머지 파라미터는 동일한 값으로 설정하여 두 시물레이션 결과를 비교하였다. $N_{ref} = 4$ 인 시물레이션은 참조노드의 배치가 그림 1과 같으며, $N_{ref} = 8$ 인 시물레이션은 그림 1의 배치로부터 실내 공간의 네 모서리의 각 중간 지점에 참조노드를 하나씩 추가한 배치이다. 모든 σ 에 대해, $N_{ref} = 8$ 로 수행한 결과가 $N_{ref} = 4$ 의 결과보다 작은 RMSE를 가지며, 2m 이하의 추정위치 오차를 보이는 격자점의 비율도 $N_{ref} = 8$ 로 수행한 결과가 더 높다. 이러한 결과는 참조노드 사이의 간격이 좁을수록 많은 격자점들이 상대적으로 정확한 $\max\{p_k\}$ 를 가지기 때문이다. 즉, 각 격자점과 가장 가까운 참조노드 사이의 거리가 작아지므로, 신호의 전파거리가 길어질수록 커지는 RSSI 측정치의 오차에 상대적으로 적은 영향을 받은 $\max\{p_k\}$ 를 토대로 A_k 를 특정하기 때문이다.

그림 7은 ML 알고리즘, VAML 알고리즘, wKNN 알고리즘의 추정위치 정확도를 비교하기 위한 그래프



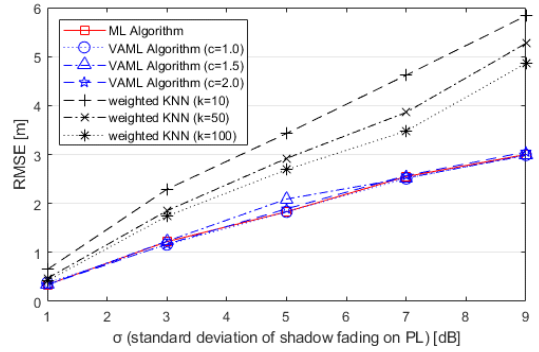
(a) RMSE by two different values of a parameter N_{ref} ($N_{ref} = 4, N_{ref} = 8$)



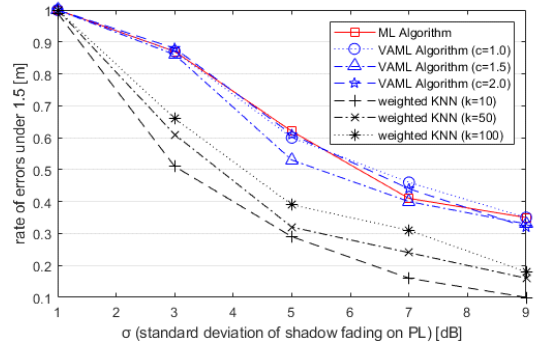
(b) Rate of errors within 2 m in estimated positions by two different values of a parameter N_{ref} ($N_{ref} = 4, N_{ref} = 8$)

그림 6. 참조노드의 수에 따른 추정위치의 정확도 비교
Fig. 6. Comparison of estimated position accuracy according to the number of reference nodes

이다. 공통적으로 $w = h = 15$, $N_{row} = N_{col} = 16$ 으로 설정하였으며, $N_{ref} = 4$ 로 참조노드의 배치는 그림 1과 같으며, $m = 5$ 또한 동일하게 설정하였다. VAML 알고리즘은 c 의 값을 1.0, 1.5, 2.0으로 설정한 세 가지 결과가 구분되도록 나타내었다. wKNN 알고리즘은 센서노드가 측정한 데이터와 가장 가까운 거리를 갖는 k 개의 labeled data를 데이터셋에서 선택하여 가까운 순으로 label에 가중치를 부여하고, voting을 통해 가장 큰 비중을 차지하는 label을 센서노드의 추정 위치로 결정하는 방식으로, 유클리드 거리를 적용하였으며 k 의 값을 10, 50, 100으로 설정한 세 가지 결과가 구분되도록 나타내었다. 그림 7(a)와 그림 7(b)를 통해, ML 알고리즘과 VAML 알고리즘이 wKNN 알고리즘에 비해 더 좋은 성능을 보였음을 알 수 있다. 또한, VAML 알고리즘은 c 의 값에 의해 추정위치의 정확도가 크게 달라지지 않았으며, ML 알고리즘의 시간 복잡도를 경감한 VAML 알고리즘의 정확도 일



(a) RMSE by ML algorithm, VAML algorithm, and wKNN algorithm



(b) Rate of errors within 1.5 m in estimated positions by ML algorithm, VAML algorithm, and wKNN algorithm

그림 7. ML 알고리즘과 VAML 알고리즘, wKNN 알고리즘의 추정위치 정확도 비교

Fig. 7. Comparison of estimated position accuracy of ML algorithm, VAML algorithm, and wKNN algorithm

화가 크지 않으므로 VAML 알고리즘이 더 좋은 효율을 보였다고 할 수 있다.

마지막으로, 본 논문에서 제시한 ML 알고리즘과 VAML 알고리즘의 시간 복잡도를 wKNN 알고리즘과 비교하여, VAML 알고리즘의 시간 복잡도가 가장 작음을 보인다.

ML 알고리즘은 모든 순서쌍 (i, j) , 즉 전체 격자점에 대해 $L(p, i, j)$ 를 계산한다. $N_{row} \cdot N_{col}$ 개의 격자점에서 식 (4)에 따라 N_{ref} 개의 $\log(f(p_k | (i, j)))$ 를 계산하는 과정의 시간 복잡도는 행의 수, 열의 수, 참조노드의 수를 모두 곱한 $N_{row} \cdot N_{col} \cdot N_{ref}$ 이고, 모든 격자점을 순회하며 $L(p, i, j)$ 가 최대인 격자점을 찾는 과정의 시간 복잡도는 $N_{row} \cdot N_{col}$ 이므로, ML 알고리즘의 시간 복잡도는 $N_{row} \cdot N_{col} \cdot (N_{ref} + 1)$ 이다. 이를 Big-O 표기법으로 나타내면 $O(N_{row} \cdot N_{col} \cdot N_{ref})$ 이다.

반면, VAML 알고리즘은 유효영역 A_k 에 포함되는 격자점들에 대해서만 $L(p, i, j)$ 를 계산하므로, A_k 에

포함되는 격자점의 수를 줄일수록 연산량과 시간 복잡도가 감소한다. A_k 에 포함된 격자점의 수를 x 라고 정의하면, A_k 에 포함된 격자점의 비율 q 를 다음과 같이 N_{row}, N_{col}, x 에 관하여 정의할 수 있다.

$$q = \frac{x}{N_{row} N_{col}}, 0 \leq q \leq 1. \quad (13)$$

VAML 알고리즘의 시간 복잡도를 산출하는 과정은 다음과 같다. 센서노드가 측정한 p 로부터 $\max\{p_k\}$ 와 $\arg \max_k\{p_k\}$ 를 구하는 과정은 측정 과정 중에 수행할 수 있으므로 시간 복잡도에 포함하지 않는다. $N_{row}N_{col}$ 개의 모든 격자점에 대해 $gridpoint_{ij}$ 를 식 (1)에 따라 구하고, R_k 와 $gridpoint_{ij}$ 의 거리 $dist(R_k, i, j)$ 를 계산한다. $dist(R_k, i, j)$ 가 $\max\{p_k\}$ 로부터 계산한 구간 $[\hat{d}_{near}, \hat{d}_{far}]$ 에 속하면 모든 $k (1 \leq k \leq N_{ref})$ 에 대한 $\log(f(p_k|(i,j)))$ 를 계산하여 더하고, 그렇지 않으면 $gridpoint_{ij}$ 에서 $L(p,i,j) = 0$ 이다. 이 과정의 시간 복잡도를 식 (13)의 q 에 대하여 나타내면 $N_{row}N_{col}(qN_{ref} + (1-q))$ 이고, 모든 격자점을 순회하며 $L(p,i,j)$ 가 최대인 격자점을 찾는 과정의 시간 복잡도는 $N_{row}N_{col}$ 이므로 VAML 알고리즘의 시간 복잡도는 $N_{row}N_{col}(qN_{ref} + (1-q) + 1)$ 이다. 이것의 Big-O 표기는 $O(qN_{row}N_{col}N_{ref})$ 또는 $O(xN_{ref})$ 이다.

wKNN 알고리즘의 시간 복잡도는 다음과 같이 산출한다. 첫째로, 데이터셋에는 $N_{row}N_{col}$ 개의 격자점에서 각각 n 개의 데이터가 있고 센서노드의 측정값은 $p \in \mathbb{R}^{N_{ref}}$ 이므로, 데이터셋에 저장된 모든 데이터와 p 사이의 거리를 계산하는 과정의 시간 복잡도는 $nN_{row}N_{col}N_{ref}$ 이다. 둘째로, 계산된 모든 거리를 정렬하는 과정의 시간 복잡도는 $nN_{row}N_{col} \log(nN_{row}N_{col})$ 이다. 셋째로, 정렬된 데이터에서 p 와 가장 가까운 k 개의 labeled data에 p 로부터의 거리가 가까운 순으로 큰 가중치를 부여하여 voting을 수행하는 과정의 시간 복잡도는 k 이다. 따라서, wKNN 알고리즘의 시간 복잡도는 앞서 설명한 과정의 시간 복잡도를 모두 더한 $nN_{row}N_{col}(N_{ref} + \log(nN_{row}N_{col})) + k$ 이고, Big-O 표기는 $O(nN_{row}N_{col}(N_{ref} + \log(nN_{row}N_{col})))$ 이다.

따라서, VAML 알고리즘이 ML 알고리즘과

표 2. 실제 파라미터 값에 대한 ML 알고리즘, VAML 알고리즘, wKNN 알고리즘의 시간 복잡도 수치 비교

($N_{row} = N_{col} = 21, N_{ref} = 4, n = 100$)

Table 2. Numerical comparison of the time complexities of the ML algorithm, VAML algorithm, and wKNN algorithm for actual parameter values

($N_{row} = N_{col} = 21, N_{ref} = 4, n = 100$)

ML	VAML		wKNN	
$N_{row}N_{col}N_{ref} + N_{row}N_{col}$	$N_{row}N_{col}(qN_{ref} + 2 - q) = xN_{ref} + 2N_{row}N_{col} - x$		$nN_{row}N_{col}N_{ref} + nN_{row}N_{col} \log(nN_{row}N_{col}) + k$	
2205	$x = 44$ ($q \approx 0.1$)	1014	$k = 10$	381229.7
	$x = 88$ ($q \approx 0.2$)	1146		
	$x = 132$ ($q \approx 0.3$)	1278		
	$x = 176$ ($q \approx 0.4$)	1410	$k = 50$	381269.7
	$x = 221$ ($q \approx 0.5$)	1545		
	$x = 265$ ($q \approx 0.6$)	1677		
	$x = 309$ ($q \approx 0.7$)	1809	$k = 100$	381319.7
	$x = 353$ ($q \approx 0.8$)	1941		
	$x = 397$ ($q \approx 0.9$)	2073		
	$x = 441$ ($q = 1.0$)	2205		

wKNN 알고리즘에 비해 작은 시간 복잡도를 가짐을 알 수 있다. 표 2는 ML 알고리즘, VAML 알고리즘, wKNN 알고리즘의 Big-O 비교가 아닌, 각각의 시간 복잡도 수식에 실제 파라미터의 값들을 대응시켜 시간 복잡도 수치를 나타낸 표이다. 시간 복잡도의 변수인 VAML 알고리즘의 x (곧 q)의 값과 wKNN 알고리즘의 k 의 값에 의한 수치가 구분되도록 나타내었다.

IV. 결 론

본 논문에서는 실내 LR-WPAN 환경을 가정하여, RSSI 핑거프린팅 기반의 두 가지 위치인식 알고리즘을 제시하였다. 첫 번째 알고리즘은 여러 참조노드로부터의 RSSI 측정치에 대한 우도를 모든 격자점에서 계산하여 최댓값을 찾는 ML 알고리즘이고, 이를 변형한 두 번째 알고리즘은 유효영역에 포함되는 일부 격자점들의 우도만을 계산하도록 시간 복잡도와 연산량을 줄인 VAML 알고리즘이다.

본 논문에서 제시한 VAML 알고리즘의 주요 파라미터로 RSSI 측정치의 평균적인 표준 편차 σ , 센서노드의 RSSI 측정 횟수 m , 참조노드의 수 N_{ref} , 추정거리의 구간 길이를 결정하는 c 등을 제시하였으며, 파라미터 값들이 제시한 알고리즘의 성능에 미치는 영향을 확인하기 위해 시뮬레이션을 수행하였다. VAML 알고리즘은 m 을 늘리거나 N_{ref} 를 늘릴수록 추정위치의 정확도가 향상됨을 확인하였다. 또한, 다양한 c 값에 대한 VAML 알고리즘의 시뮬레이션 결과를 비교함으로써, ML 알고리즘에 비해 추정위치 정확도의 열화가 크지 않으며 wKNN 알고리즘에 비해 추정위치의 정확도가 높음을 확인하였다. 마지막으로, 제시한 두 알고리즘과 wKNN 알고리즘의 시간 복잡도를 비교하여 검증하였다.

본 논문을 통해 제시한 VAML 알고리즘은 센서노드가 측정한 RSSI 값으로부터 간단한 계산으로 위치를 추정하는 ML 알고리즘의 시간 복잡도를 더욱 낮추어, 실내 위치인식 시스템의 하드웨어 요구사항 및 컴퓨팅 비용을 줄이는 데에 목적이 있다. 따라서, 저가의 장치를 활용하는 실내 LR-WPAN 등의 환경에서 위치인식 시스템의 실용성을 제고할 수 있을 것으로 기대된다.

References

[1] D. Dardari, M. Luise, and E. Falletti, *Satellite and Terrestrial Radio Positioning Techniques: A Signal Processing Perspective*, 1st Ed., Academic Press, 2016.

[2] S. Bottigliero, D. Milanese, M. Saccani, and R. Maggiore, "A low-cost indoor real-time locating system based on TDOA estimation of UWB pulse sequences," *IEEE Trans. Instrumentation and Measurement*, vol. 70, pp. 1-11, Mar. 2021. (<http://doi.org/10.1109/TIM.2021.3069486>)

[3] H. Yen, L. Yang, and Z. Tsai, "3-D indoor localization and identification through RSSI-based angle of arrival estimation with real wi-fi signals," *Trans. Microw. Theory and Techniques*, vol. 70, no. 10, pp. 4511-4527, Oct. 2022. (<http://doi.org/10.1109/TMTT.2022.3194563>)

[4] Y. Kim and D. Han, "Indoor location

classification algorithm using RSSI and device height," *J. KICS*, vol. 42, no. 8, pp. 1573-1580, Aug. 2017.

(<http://doi.org/10.7840/kics.2017.42.8.1573>)

[5] S. Tomic, M. Beko, and R. Dinis, "3-D target localization in wireless sensor networks using RSS and AoA measurements," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3197-3210, Apr. 2017. (<http://doi.org/10.1109/TVT.2016.2589923>)

[6] A. Coluccia and A. Fascista, "On the hybrid TOA/RSS range estimation in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 361-371, Jan. 2018. (<http://doi.org/10.1109/TWC.2017.2766628>)

[7] N. Singh, S. Choe, and R. Punmiya, "Machine learning based indoor localization using wi-fi RSSI fingerprints: An overview," *IEEE Access*, vol. 9, pp. 127150-127174, Sep. 2021. (<http://doi.org/10.1109/ACCESS.2021.3111083>)

[8] A. Poullose and D. S. Han, "Performance analysis of fingerprint matching algorithms for indoor localization," *2020 ICAIIC*, pp. 661-665, Fukuoka, Japan, Feb. 2020. (<http://doi.org/10.1109/ICAIIC48513.2020.9065220>)

[9] K. Chintalapudi, A. Iyer, and V. Padmanabhan, "Indoor localization without the pain," in *Proc. Sixteenth Annu. Int. Conf. MobiCom 2010*, pp. 173-184, New York, USA, Sep. 2010. (<http://doi.org/10.1145/1859995.1860016>)

[10] G. Huang, Z. Hu, J. Wu, H. Xiao, and F. Zhang, "WiFi and vision-integrated fingerprint for smartphone-based self-localization in public indoor scenes," *IEEE Internet of Things J.*, vol. 7, no. 8, pp. 6748-6761, Aug. 2020. (<http://doi.org/10.1109/JIOT.2020.2974928>)

[11] C. Zhou, J. Liu, M. Sheng, Y. Zheng, and J. Li, "Exploiting fingerprint correlation for fingerprint-based indoor localization: A deep learning based approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5762-5774, Jun. 2021. (<http://doi.org/10.1109/TVT.2021.3075539>)

[12] IEEE, 802.15.4a-2007, “*IEEE Standard for Information Technology—Local and 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): Amendment 1: Add Alternate PHYs*,” 2007. (<http://doi.org/10.1109/IEEESTD.2007.4299496>)

배 종 현 (Jonghyeon Bae)



2024년 2월: 경북대학교 컴퓨터학부 학사
2024년 3월~현재: 경북대학교 컴퓨터학부 석사과정
<관심분야> 5G/6G, Tactical networks, Networked PNT (Positioning, navigation and timing), Manned-Unmanned Teaming

백 호 기 (Hoki Baek)



2006년 2월: 아주대학교 정보 및 컴퓨터공학 학사
2008년 2월: 아주대학교 정보통신공학 석사
2014년 2월: 아주대학교 컴퓨터공학 박사
2014년 3월~2015년 2월: 아주대학교 장위국방연구소 전임연구원
2015년 3월~2021년 2월: 아주대학교 국방디지털융합학과 강의교수
2021년 3월~현재: 경북대학교 컴퓨터학부 조교수
<관심분야> 5G/6G, Spectrum sharing, Non-terrestrial networks, UAV networks, Tactical networks, IoT, V2X, Networked PNT (Positioning, navigation and timing), Manned-Unmanned Teaming
[ORCID:0000-0001-9213-7845]