

A Moving Terminal's Coordinates Prediction Algorithm and an IoT Application

Daewon Kim*

Abstract

Recently in the area of ICT, the M2M and IoT are in the spotlight as a cutting edge technology with the help of advancement of internet. Among those fields, the smart home is the closest area to our daily lives. Smart home has the purpose to lead a user more convenient living in the house with WLAN (Wireless Local Area Network) or other short-range communication environments using automated appliances. With an arrival of the age of IoT, this can be described as one axis of a variety of applications as for the M2H (Machine to Home) field in M2M. In this paper, we propose a novel technique for estimating the location of a terminal that freely move within a specified area using the RSSI (Received Signal Strength Indication) in the WLAN environment. In order to perform the location estimation, the Fingerprint and KNN methods are utilized and the LMS with the gradient descent method and the proposed algorithm are also used through the error correction functions for locating the real-time position of a moving user who is keeping a smart terminal. From the estimated location, the nearest fixed devices which are general electric appliances were supposed to work appropriately for self-operating of virtual smart home. Through the experiments, connection and operation success rate, and the performance results are analyzed, presenting the verification results.

▶ Keyword : M2M, IoT, Smart Home, WLAN, Fingerprint, LMS, M2H, Gradient Descent Algorithm

I. Introduction

최근 인터넷 보급의 확산과 스마트 기기의 발전으로 인하여 무수히 많고도 다양한 기기를 하나의 네트워크를 통해 제어하고 활용하는 시대가 되었다. 최근 모바일 인터넷의 발전으로 언제 어디서나 사용이 가능한 유비쿼터스 컴퓨팅 및 네트워크라는 용어가 많이 사용되었는데 현재 그리고 앞으로는 이 유비쿼터스의 개념을 넘어선 사물인터넷 (Internet of Things), 그리고 만물인터넷 (Internet of Everything)의 시대가 도래 하게 될 것으로 전망되고 있다. 사물인터넷은 M2M (Machine to Machine)과 IoT (Internet of Things) 이 두 가지로 대표 되는데 M2M은 말 그대로 기계와 기계간의 통신을 의미하는 것이고 IoT는 사람을 포함하여 모든 사물이 인터넷을 통해 네트워

크를 형성 한다는 정의이다 [1][2]. 사물인터넷은 우리 생활의 여러 가지 분야에 적용 시킬 수 있는데, 그 중의 하나가 바로 M2H (Machine to Home), 다시 말해서 스마트 홈 (Smart Home) 분야이다. 스마트 홈이란 사람이 가장 편안하게 휴식을 취하는 공간인 집에서 디바이스와 시스템이 자동으로 제어되어 사용자에게 높은 효율성과 편리함을 제공하는 환경을 뜻한다. 최근 기존의 스마트 홈에 유비쿼터스 개념과 사물인터넷 개념을 적용시켜 새로운 개념의 스마트 홈을 구현하고자 하는 연구들이 시도되고 있다 [3][4]. 최종적으로는 집안 내의 모든 가전을 컴퓨터화 시켜서 사물들끼리 네트워크로 연결된 하나의 커뮤니티를 구성하고 이들이 사용자에게 서비스를 자동으로 제

• First Author: Daewon Kim, Corresponding Author: Daewon Kim

*Daewon Kim (drdwkim@dgu.edu), Dept. of Applied Computer Engineering, Dankook University

• Received: 2017. 02. 02, Revised: 2017. 05. 10, Accepted: 2017. 06. 26.

공하도록 하는 것이 목적이다. 사용자는 이러한 가전 기기들을 직접적으로 동작시키지 않고 거기에 장착된 컴퓨터를 통해 원격 제어 또는 상황에 따라서 사물지능 통신을 통해 스스로 동작하게 한다 [5][6]. 본 연구에서는 IoT와 M2H 개념을 활용하여 효율적이고 능동적인 스마트 홈 시스템 구현을 도모하고 스마트 홈 네트워크 구축을 진행하였다. 사용자가 스마트 홈 내에서 단말기를 작동 시키는 방법은 두 가지가 있는데, 하나는 리모트 컨트롤러처럼 무선 환경에서 원격 조종하는 방식이 있고, 다른 하나는 어떤 특정 상황에 직면하여 사물 스스로 적응하여 적절히 동작하는 방법이다. 이 중에서 후자의 경우에는 다양한 방법이 존재한다. 우리가 흔히 접하는 출입구 자동문이나 현관의 전등 같은 경우가 이에 해당 되는데 기존 방식은 이 기기에 센서를 부착한 후 움직이거나 이동하는 사람을 감지하여 작동하는 센서 네트워크의 개념으로 사용 되었다. 이 방식은 모든 사물에 센서를 부착해야 하는 번거로움과 현재 On/Off 기능 정도만 사용 되고 있기 때문에 다양한 사물에 접목시키기가 어려운 측면이 존재한다. 이에 따라, IoT의 M2H 개념 구현을 위하여 WLAN (Wireless Local Area Network) 이라는 무선 LAN 환경을 활용할 수 있는데 이 기술은 어느 곳에서나 접속이 가능하다는 장점이 있다 [7][8]. LAN 카드만 부착 되어있다면 어떠한 임베디드 장비도 네트워크를 통하여 쉽게 접속이 가능하고 상황에 맞도록 적절히 동작하는 것이 가능하다. 스마트 홈의 조건이 갖추어진 환경에서는 일반 사용자가 누릴 수 있는 편안함이 많다. 예를 들어, 외부 온도에 맞는 냉난방 장치 가동, 간단한 제스처를 통한 가전기기 제어, 출입만으로 통제 가능한 현관 door-lock 등이 그것이다. 본 논문에서는 WLAN 환경에서 사용자의 정확한 위치와, 사물과 사용자의 거리를 파악하여 실내공간에서의 적절한 상황에 맞도록 효율적인 스마트 홈 시스템을 구축하고 향후 사물인터넷 및 만물인터넷 시대에 어울리는 활용 가능성을 연구하고 평가하였다 [9][10]. 본 논문에서는 스마트 홈 내의 무선 환경에서 사용자의 위치 인식을 통한 홈 서비스를 구축하고 제공하는 것과 현재 진행되고 있는 표준화 내용과 연계하여 응용 가능한 부분을 찾아 적절히 활용할 수 있는 방법을 연구하였다. 따라서 본 연구에서는 M2M과 IoT 개념을 활용하여 M2H를 적용한 스마트 홈 환경을 구현하고 직접 임베디드 환경을 통하여 구축해 보았다.

II. Location Estimation Algorithm

일반 사용자의 위치를 추적하는 기존 기술로는 GPS와 RFID, 초음파 방식 등이 있다. 이러한 기술들은 실외 지역이나 그 외 광범위한 영역에서 장비와 이들의 설치에 관한 비용 등의 이유로 인한 단점이 존재한다. 그런데 WLAN (Wireless Local Area Network) 방식은 지금 현재 필요한 많은 곳에 설치가 이미 잘 되어있고 쉬운 접근성을 지니고 있다. 또한 별도

의 장비가 필요 없이 AP (Access Point) 만을 사용하기 때문에 적은 비용으로 효율적인 위치 추정을 이용한 사물인터넷 시스템 구축이 가능하다. 무선 랜 환경에서 위치를 추정하는 방식으로 많은 연구가 진행 되었는데 가장 대표적인 측정 방식으로는 삼각측량법과 Fingerprint, K-Nearest Neighbor 방식 등이 있다.

1. Triangulation Method

삼각측량법은 그림 1과 같이 세 기준점의 좌표와 측정하고자 하는 모바일 장치간의 거리를 이용하여 현재의 위치를 측정하는 방식이다. 그림 1에서의 기준점 A, B, C는 고정된 노드로서 WLAN 환경에서는 AP로 표현 할 수 있다. 기준점과 모바일 장치간의 거리를 D 라 할 때 기준점과 모바일 장치간의 거리를 바탕으로 3개의 원이 생성 되는데 이 원들의 교차점 $P(x,y)$ 가 모바일 장치의 위치라고 정의 할 수 있다. 기준점과 모바일 장치의 좌표의 거리는 수 식 (1), (2), (3)과 같이 나타 낼 수 있는데, 이를 이용하여 교차점 P 를 구할 수 있다. D_A, D_B, D_C 는 수신된 전파의 경로 손실을 이용하여 얻은 거리라 볼 수 있는데 이를 구하는 방법으로는 수신 신호세기를 뜻하는 RSSI (Received Signal Strength Indication) 정보를 이용하는 경우와 도착시간인 TOA (Time of Arrival) 정보를 이용하는 경우가 있으며 일반적으로는 RSSI 값을 이용한다.

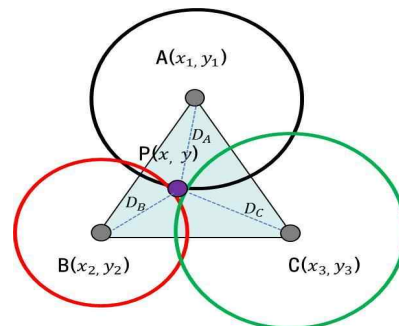


Fig. 1. Triangulation Method

$$D_A = \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \quad (1)$$

$$D_B = \sqrt{(x_2 - x)^2 + (y_2 - y)^2} \quad (2)$$

$$D_C = \sqrt{(x_3 - x)^2 + (y_3 - y)^2} \quad (3)$$

여기서 RSSI는 아래 수 식 (4)에 제공된 공식을 통해 알 수 있다.

$$L = 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right) [dB] \quad (4)$$

수 식 (4)에서 L 은 송신기와 수신기 사이의 경로 손실로서 RSSI 값을 나타내고, λ 는 전파의 파장이며 d 와 같이 표현 할 수 있다. 수 식 (4)에서 d 를 표현하면 수 식 (5)와 같다.

$$d = \frac{\lambda}{4\pi} \times 10^{\frac{L}{20}} = \frac{c}{4\pi f} \times 10^{\frac{L}{20}} \quad (5)$$

수 식 (5)에서 c 는 전파속도이고 f 는 주파수를 나타낸다. 일반적인 경우 주파수는 2.4GHz 대역을 사용하며 공기 중 무선 신호의 전파 속도는 $3 \times 10^8 [m/s]$ 이다.

2. Fingerprint Method

Fingerprint 기법은 WLAN 환경의 실내 위치 추정에서 가장 많이 쓰이는 방식으로, 실내 지역에서 참조 위치 (RP: Reference Point)를 선정하여 각 AP에서부터 측정되는 신호세기 정보인 RSSI를 수집하고 이를 바탕으로 모바일 장치의 신호정보를 통해 위치를 추정 하는 방식이다. 그림 2와 같이 참조 위치인 RP를 정하여 신호 세기, 즉 RSSI를 데이터베이스화 하는 단계를 준비 단계(정보 수집 단계)라 하고 모바일 장치의 신호 정보를 이용하여 최종 위치를 추정하는 단계를 수행 단계 (위치 추정 단계)라 한다 [11][12].

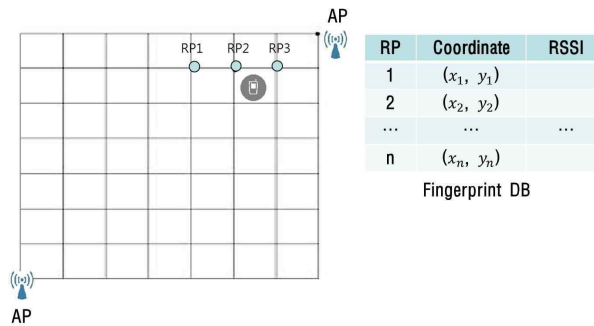


Fig. 2. Fingerprint Method

그림 3은 Fingerprint 과정을 준비 단계와 수행 단계인 두 단계로 나눈 순서도이다. 준비 단계에서 각각의 위치에 대한 신호세기를 측정한 후 수행 단계에서 위치 결정 알고리즘을 사용하여 정확한 위치를 결정해야 하는데 이를 위해서는 이전 단계인 준비 단계에서 정확한 데이터베이스를 구축해야 하고 정확성과 효율성이 뛰어난 위치 결정 알고리즘을 사용해야 한다.

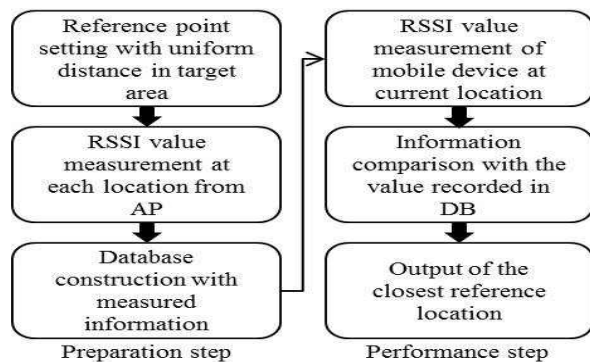


Fig. 3. Flowchart of Fingerprint Method

많이 쓰이는 위치 결정 알고리즘으로는 인공신경망을 사용하는 Neural Network (NN) method, KNN(K-Nearest Neighbor) method, 그리고 Bayesian 분포법 등 다양한 방법들이 있으며 이 알고리즘들은 모두 machine learning 분야인 기계학습이나 패턴인식 (pattern recognition) 같은 분야에서 응용 되어 많이 사용되고 있다. 본 논문에서는 위치 측정 분야에서 사용될 수 있는 다양한 알고리즘들 중 몇 개의 근접 노드를 통한 최적의 선택 기법으로 알려진 KNN 알고리즘을 이용해서 초기 위치를 추정하고 이를 보정하는 단계를 포함하고자 오류 최소화 알고리즘인 LMS (Least Mean Square) 기법을 이용하여 최적의 위치를 찾아내기 위해 적정 위치 파라미터를 적용하는 방법을 제안하였다.

3. K-Nearest Neighbor Algorithm

KNN은 패턴 인식에 많이 활용되는 알고리즘으로 높은 정확도와 낮은 오류로 추론 영역에서 많이 사용 되는 알고리즘이다. 일반적으로 KNN 알고리즘은 분류가 되어 있지 않은 데이터인 test data와 분류된 데이터인 training data를 이용하여 test data를 training data 중에서 가장 비슷한 속성을 가진 그룹으로 분류해주는 기능을 제공한다. 이 때 test data와 training data 간의 거리를 측정하여 유사도를 결정하는데 그 결정 방식으로는 유클리디안 거리 측정 기법과 맨해튼 거리 측정 방식이 사용된다. KNN 알고리즘은 간단하고 효과적이며 학습과정이 빠르다는 장점이 있지만 모든 데이터들 간의 거리를 계산해야 되기 때문에 데이터의 크기가 커지면 비용이 동시에 많이 발생한다는 단점이 존재한다. 위치 추정 방식을 위한 Fingerprint 방식에서 KNN 알고리즘은 먼저 설치된 AP로부터 단말기에 수신되는 신호를 측정한 후 해당 위치에서의 신호 값과 데이터베이스에 저장되어있는 참조 위치에서의 신호 값을 비교하고 분석하여 식 (6)과 같은 유사도를 측정 할 수 있다 [13][14].

$$D_i = \sum_{j=1}^m |S_j - s_j|, i = 1, 2, \dots, n \quad (6)$$

수 식 (6)에서 D_i 는 임의의 현재 위치 $P(x, y)$ 에서 m 개의 AP로부터 측정된 신호 값 $(S_1, S_2, \dots, S_m)$ 과 Fingerprint 데이터베이스에 저장된 i 번째의 참조 위치에서 AP로부터 측정된 신호 값 $(s_1, s_2, \dots, s_m)$ 을 비교하여 얻어진다. 그 후 위치 $P(x, y)$ 에서의 유사도 D_i 는 참조 위치의 개수에 해당하는 n 개가 형성 된다. 그리하여 계산된 D_i 를 이용하여 얻어진 유사도 중에서 가장 낮은 K 개의 값을 지정한다. K 값을 결정하기 위해서는 일정한 데이터의 양이 필요한데 이를 위해 D 를 통한 MSE (Mean Square Error)를 사용하고 그 식은 (7)과 같다. K 의 양에 따라 거리 좌표들의 오차율을 이용하여 K 값을 결정하게 되고 결정된 K 값과 수 식 (8)을 통해

현재의 위치인 $P(x, y)$ 를 추정 한다.

$$E = \sqrt{(x - x_1)^2 + (y - y_1)^2} \quad (7)$$

$$(\hat{x}, \hat{y}) = \frac{1}{K} \sum_{i=1}^K (x_i, y_i) \quad (8)$$

수 식 (7)에서 계산된 E 는 오차율을 나타내는데 오차율이 작으면 작을수록 정확도가 높은 것이고 이는 가장 적절한 K 를 선택 할 수 있게 한다. 수 식 (8)에서의 i 는 n 개의 참조 위치 중 선택된 K 개의 참조 위치들이고 $(\hat{x}, \hat{y})$ 는 현재 위치 $P(x, y)$ 와 가장 유사한 추정 위치이다. 삼각측량법, 핑거프린트 기법, 그리고 KNN 알고리즘 이외에도 표 1에 다양한 위치 추정 알고리즘 기법들의 상호 장단점 및 특징이 비교 평가 되어 있다. [15]

Table 1. Comparison of Various Location Algorithms

Algorithm	Methods	Features	Merits/Shortcomings
Cricket	Ultrasonic, RF	Location Estimation w/ a receiver	Extensible Energy problem
Active Badge	IrDA	Badge and receiver use	Simple system
RADAR	Wireless LAN	IEEE 802.11 embedded & compatible	Easy use Big Equipment
Easy Living	Scene Analysis	High Technology Camera	Expensive High location accuracy

III. Moving Terminal's Location Estimation Algorithm

1. LMS Algorithm

LMS (Least Mean Square Algorithm) 알고리즘은 학습 하는 동안 출력 층의 활성화 함수로 항등함수가 사용되고, 학습규칙을 이용하여 출력 층의 출력 값과 목표 값과의 차이를 최소화 하는 것이다 [16][17][18]. 이 알고리즘은 입력 값에 따라서 필터를 적용하여 나온 출력 값과 예상된 값의 차이를 수치로 표현 할 수 있는데 오류에 대한 수 식은 다음과 같다.

$$E = d - W(X) \quad (9)$$

여기서 d 는 예상된 출력 값이고 $W(X)$ 는 현재 출력 값이 된다. 현재 출력 값이 1차 함수라면 오차는 2차 함수가 되기 때문에 기울기가 0일 때 오차는 최소가 된다. 그림 4에서 최소의 오차인 E 를 가지는 위치가 최적의 W 가 되고 그 수 식은 (10)과 같이 표현이 가능하다.

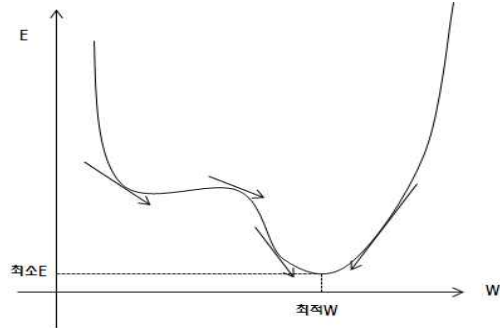


Fig. 4. Gradient Descent Method

$$W_{n+1} = W_n - \mu \frac{\partial E}{\partial W} \quad (10)$$

최소의 오차점을 찾기 위해서는 W 에 관하여 오차 함수식을 편미분하여 기울기의 크기를 측정한다. 기울기를 값에 따라 빼고 더하는 과정을 반복하면 최소가 되는 점에 도달할 수 있는데 이러한 방식으로 최적의 값을 찾아가는 방법을 경사하강법 (Gradient Descent Method) 이라 한다 [19][20]. 수 식 (10)에서 μ 는 가중치를 갱신하는 간격을 조절하는 상수로서 학습률 (Learning Rate)에 해당한다. μ 를 너무 작게 설정하면 최소점에 도달할 때까지 많은 시간이 소요되고, 너무 큰 값을 설정하면 최소점에 수렴하지 못할 수도 있기 때문에 일반적으로 0보다 크고 1보다 작은 양의 값을 설정한다.

2. Proposed Method for Location Estimation of a Moving Target

본 연구를 통해 제안된 위치 측정 알고리즘에서는 사물인터넷 분야에서도 M2H 분야에의 적용을 위해 해당 파라미터들을 설정하였다. 일반적인 위치 추정 분야에서의 좌표 값을 대입하여 나타낸 모양은 그림 5와 같다. Q가 실제 위치라 할 때 P는 근접하게 측정된 추정 위치이다. R_1 부터 R_4 는 각각 참조 위치이고 이것은 수 식 (10)에 있는 E 를 구하기 위해 활용된다. E 는 각각의 좌표에서의 거리를 이용한 MSE 값을 이용하여 구할 수 있는데 1차로 결정된 추정 위치 $P(x_p, y_p)$ 와 각 참조 위치 R_1 부터 R_4 사이의 거리 값을 계산하여 오차를 구하는 식을 만든다.

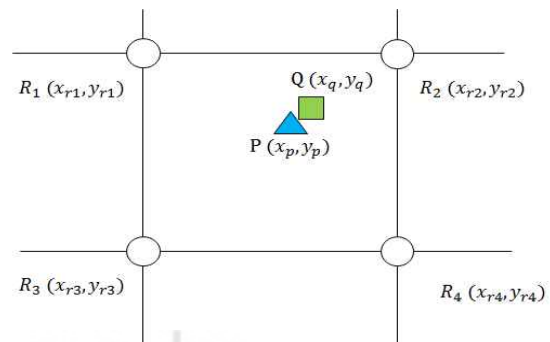


Fig. 5. Estimation (P) of Q using Reference Points R

$$E = \frac{1}{N} \sum_{i=1}^N (D_i - D_{mi})^2 \quad (11)$$

$$= \frac{1}{N} [(D_1 - D_{m1})^2 + \dots + (D_N - D_{mN})^2]$$

$$D_{mi} = \sqrt{(x_{ri} - x_p)^2 + (y_{ri} - y_p)^2} \quad (12)$$

$$D_i = \sqrt{(x_{ri} - x_q)^2 + (y_{ri} - y_q)^2} \quad (13)$$

여기서 D_{mi} 는 최대 근접 위치로 추정된 위치와 참조 위치 간의 거리이고 D_i 는 E 의 수 식 구성을 위해 논리적으로 알고 있는 실제 위치와 참조 위치 간의 거리이다. 위치 측정을 위해 이와 같이 만든 식은 수 식 (10)에 나타난 바와 같이 연산이 이루어지는데 W 는 각각 x 와 y 좌표가 된다. 이렇게 되면 수 식 (10)을 수 식 (14)와 같이 다시 전개하도록 한다.

$$W(x, y)_{n+1} = W(x, y)_n - \mu \frac{\partial E(x, y)}{\partial W(x, y)} \quad (14)$$

수 식 (14) 내에 있는 $\frac{\partial E(x, y)}{\partial W(x, y)}$ 을 먼저 x_q 에 대하여 전개하면 수식 (15)와 같이 나타낼 수 있고, 이때 N 은 그림 5를 보면 참조 위치가 네 개 이므로 4가 된다.

$$\begin{aligned} \frac{\partial E}{\partial x_q} &= \frac{\partial}{\partial x_q} \left(\frac{1}{N} \sum_{i=1}^N (D_i - D_{mi})^2 \right) \quad (15) \\ &= \frac{\partial}{\partial x_q} \left(\frac{1}{N} \sum_{i=1}^N \left(\sqrt{(x_{ri} - x_q)^2 + (y_{ri} - y_q)^2} \right. \right. \\ &\quad \left. \left. - \sqrt{(x_{ri} - x_p)^2 + (y_{ri} - y_p)^2} \right) \right) \end{aligned}$$

수 식 (15)를 y_q 에 대해서도 전개하면 수 식 (16)과 같이 나타낼 수 있다.

$$\begin{aligned} \frac{\partial E}{\partial y_q} &= \frac{\partial}{\partial y_q} \left(\frac{1}{N} \sum_{i=1}^N (D_i - D_{mi})^2 \right) \quad (16) \\ &= \frac{\partial}{\partial y_q} \left(\frac{1}{N} \sum_{i=1}^N \left(\sqrt{(x_{ri} - x_q)^2 + (y_{ri} - y_q)^2} \right. \right. \\ &\quad \left. \left. - \sqrt{(x_{ri} - x_p)^2 + (y_{ri} - y_p)^2} \right) \right) \end{aligned}$$

$$\begin{aligned} x_{n+1} &= x_n - \mu \frac{\partial E(x, y)}{\partial x_q} \\ &= x_n - \mu \frac{E|x = x_n + \Delta x_n - E|x = x_n}{|y = y_n \quad |y = y_n} \Delta x_n \\ &= x_n - \mu \frac{\Delta E}{\Delta x_n} \end{aligned}$$

수 식 (16)에서도 수 식 (15)와 마찬가지로 N 은 4가 된다. 이리하여 수 식 (14)를 전개하면 x 와 y 에 관하여 수 식 (17)과 수 식 (18)처럼 풀 수 있다. 수 식 (11)에 나타난 오차 E 를 각각 x_q 와 y_q 에 대하여 편미분한 결과가 수 식 (17)과 (18)에 나타나 있다.

$$\begin{aligned} y_{n+1} &= y_n - \mu \frac{\partial E(x, y)}{\partial y_q} \quad (18) \\ &= y_n - \mu \frac{E|x = x_n + \Delta x_n - E|x = x_n}{|y = y_n + \Delta y \quad |y = y_n} \Delta y_n \end{aligned}$$

이 식에 의해서 알고리즘은 수행되고 어느 특정 조건에서 반복 연산을 멈추게 되는데 그 값을 임계치라 하고 수 식 (19)와 같이 표현이 가능하다.

$$l = |x_n - x_{n-1}|, m = |y_n - y_{n-1}| \quad (19)$$

식 (19)에서 l 과 m 은 각각 x 와 y 에 대한 임계치를 표현한 값으로 이 두 값이 사전에 지정된 특정 조건을 동시에 만족해야 수행이 중지된다.

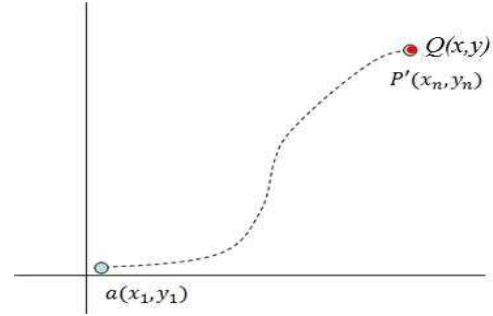


Fig. 6. Estimated Location (P') Starting from a.

초기값 $a(x_1, y_1)$ 이 0에 수렴하는 작은 수에서 출발하여 제안된 알고리즘이 수행되다가 최종적으로 실제 위치인 점 Q 를 찾아가게 되는데 Q 의 도달을 지향하며 수 식 (19)의 조건을 만족하는 그림 6에서의 $P'(x_n, y_n)$ 이 본 연구에서 제안한 알고리즘을 통한 최종적인 추정 위치가 된다.

IV. Experiment and Results

1. Experiment of Location Estimation

그림 7에 본 연구를 위해 진행한 실험의 흐름도가 나타나 있다. 먼저 실험 장소에서 각 참조 위치별로 신호의 세기를 측정하여 데이터베이스에 저장한 후 임의의 지점에서 모바일 장치로 신호의 세기를 측정하여 비교한 뒤 KNN 방식으로 현재의 위치를 추정한다. 이어서 새롭게 측정된 위치 좌표를 제안한 알고리즘을 통한 좌표 보정을 통해 최종 위치 값으로 도출한 뒤 실제 위치와 보정 전 후 좌표를 비교 분석한다. 그리고 스마트 홈이라는 상황을 가정하고 지정된 위치에 각각 고정된 4개의 기기와의 통신을 통하여 이동노드의 현재 위치와 고정 기기들 간의 거리를 계산하여 기기들을 동작시키는 방식을 통해 휴대 기기로 장치를 제어하는 스마트 홈 시스템을 구축해 보았다.

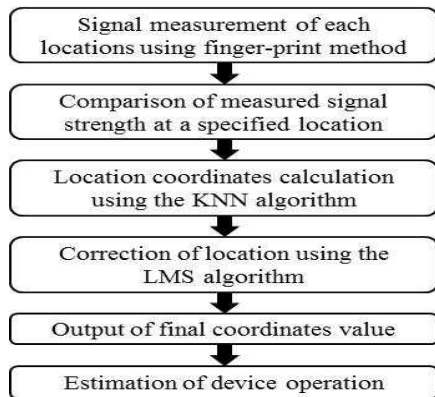


Fig. 7. Flowchart of Experiments

Table 2. Experiment Places and Used Devices

Place	General Experimental Laboratory
Size	10m × 20m × 4m
AP	Four IEEE 802.11n compatible devices (Four Antennas for each Ap) (Maximum Antenna Sensitivity = 5dBi)
# of RP	9 or 12
Moving Node	Smart Phone Terminal (SM-N910G) (Height = 1m)
Fixed Device	Four Embedded board (Raspberry PI)

본 연구에서 진행한 실험의 장소와 환경에 관한 정보는 표 2에 나타나 있다.

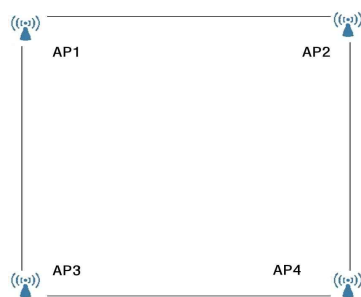
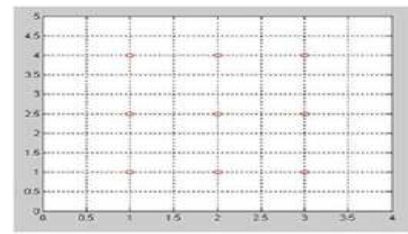
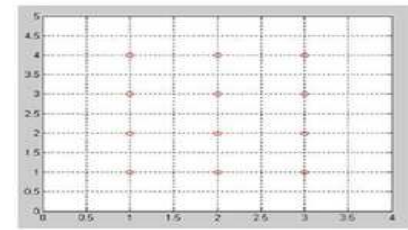


Fig. 8. Experimental Setup & AP Location (Height=1m)

실험 장소로 쓰인 곳은 가로 약 10m, 세로 약 20m, 높이 약 4m 크기의 대형 종합 실험실이다. AP는 802.11n을 기반으로 하는 동일한 성능의 기기로서 그림 8과 같이 실내의 네 구석 모두 높이 1m의 위치에 각각 설치하였다. 그림 9는 참조 위치인 RP를 실험 장소를 기준으로 하여 그래프로 나타낸 것인데 그래프 상에서 가로는 좌표 한 칸 당 약 2.5m, 세로 좌표는 한 칸 당 약 4m의 거리차가 존재한다. 그래프 내에 설정된 RP는 임의의 점 $Q(x, y)$ 를 추정할 때 기준으로 삼을 수 있는 참조 점을 나타낸 것이다. RP는 그림 9처럼 3×3 과 3×4 두 가지 경우를 고려하여 실험을 진행 하였다.

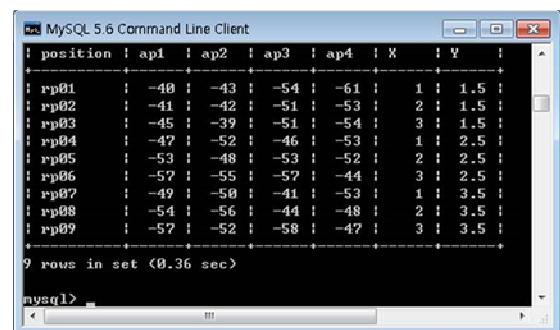


(a) Location coordinates of RP=9

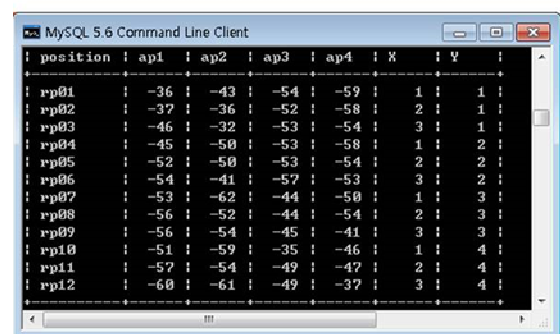


(b) Location coordinates of RP=12

Fig. 9. Coordinates of Reference Points



(a) DB value when RP=9



(b) DB value when RP=12

Fig. 10. RSSI Values Received from AP

그림 10과 그림 11은 거리와 신호세기의 상관관계를 알아보기 위하여 각각의 AP로부터 임의로 설정해둔 RP에서의 신호세기를 데이터베이스화 시키고 색으로 표현한 것인데 RGB 값에 따라 신호세기가 강할수록 붉은색 계열로 나타나고 신호세기가 약할수록 푸른색 계열로 나타나게 되어 AP로부터 가까울수록 신호세기가 강해지고 멀어질수록 더욱 더 약해진 신호세기가 측정되는 것을 알 수 있다. 그림 11의 (a)는 전체 위치에서 그림 8에 나타난 1번 AP로부터 수신되는 신호를 나타낸 분포도이고 (b), (c), (d)는 각각 그림 8의 2, 3, 4번 AP로부터 수신되는 신호를 나타낸 분포도이다.

네 가지 그림 모두 AP로부터 멀어질수록 푸른색 계열의 색으로 표현되어 지고 있음을 알 수 있다. 따라서 AP로부터 위치가 멀어질수록 신호의 세기는 수신감도가 약해짐을 알 수 있다. 실험 시 모바일 장치의 신호와 핑거프린트 DB에 기록되어 있는 각 RP에서의 신호세기를 이용하여 위치를 추정 하는데 이 때 KNN 방식을 사용한다. KNN은 학습 데이터가 소량인 경우 K 의 값을 정하기 어려울 뿐만 아니라 결과값의 오차율도 커지고 또한 신뢰도가 떨어지므로 세심한 고려가 필요하다.

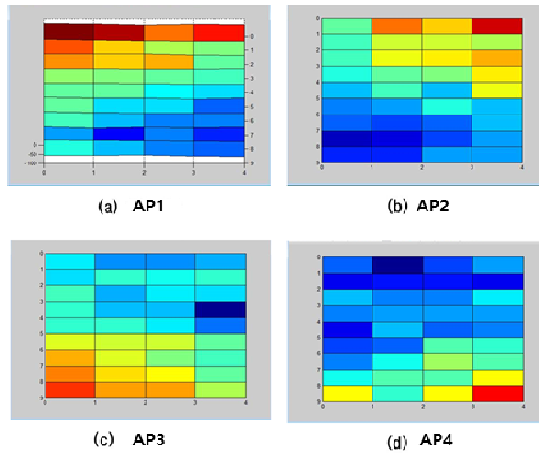


Fig. 11. Strength Map of RSSI Values Received from Four APs

본 연구에서 제안한 방법으로는 학습 데이터들을 RP의 위치 좌표로 설정하였는데 이 경우에 서로 다른 위치에서 측정을 하여도 똑같은 위치 좌표가 존재 하게 된다. 따라서 이러한 단점을 보완하기 위해서 좌표를 획득하기 위한 새로운 참조 위치인 TP (Test Point)를 사용하였다. 그림 12에 RP와 함께 5×10 크기의 새로운 참조 위치인 TP가 나타나 있다. 이렇게 하여 좀 더 세밀한 후보영역의 좌표 값을 도출 하도록 하였다.

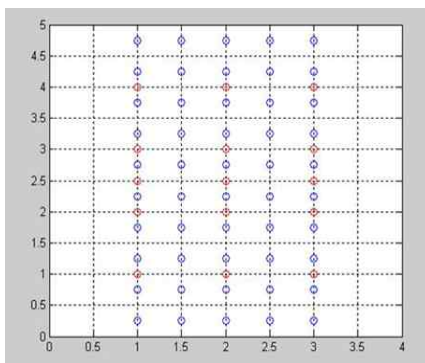


Fig. 12. Coordinates Map of TP (Blue) and RP (Red)

그 다음 TP와 현재 위치 $Q(x,y)$ 와의 유사도를 비교하여 K 의 개수와 최종 위치를 추정한다. 표 3은 총 20회의 실험을 통하여 참조 위치에서 K (9, 10, 11, 12)와 RP (9, 12)의 개수를 달리하여 KNN과 본 연구에서 제안한 기법을 이용하여 추정

된 위치를 정리한 결과이다.

Table 3. Estimated RP Number Results from the KNN and the Proposed Methods ($K=9,10,11,12$ & $RP=9,12$)

#	RP	RP No.	K=9		K=10		K=11		K=12	
			Est. RP		Est. RP		Est. RP		Est. RP	
			KNN	Proposed	KNN	Proposed	KNN	Proposed	KNN	Proposed
1	9	7	7	7	7	6	8	7	7	7
	12	6	7	6	6	6	6	6	6	6
2	9	6	5	6	6	6	6	7	6	6
	12	11	11	11	11	11	10	11	11	11
3	9	1	1	1	1	1	1	1	1	1
	12	7	7	7	8	7	7	7	7	7
4	9	8	8	7	6	8	8	8	8	9
	12	2	2	2	2	1	2	2	1	2
5	9	5	5	5	5	5	5	5	5	5
	12	1	2	1	1	1	1	1	1	1
6	9	7	8	7	6	7	7	7	7	7
	12	5	5	5	5	5	5	5	5	5
7	9	2	2	2	2	2	2	2	2	2
	12	12	12	10	10	12	12	12	12	12
8	9	4	4	4	1	4	4	4	4	4
	12	8	8	8	8	8	9	8	8	8
9	9	3	3	3	3	3	4	3	3	3
	12	9	10	9	9	9	9	9	7	9
10	9	9	9	9	9	9	9	9	7	9
	12	10	10	10	9	10	9	10	10	10
11	9	9	6	8	9	9	9	9	9	9
	12	1	1	1	1	1	1	1	1	1
12	9	1	1	1	1	1	1	1	1	1
	12	3	3	3	3	3	5	2	3	3
13	9	7	7	7	7	7	7	7	7	7
	12	7	7	7	7	6	7	7	6	7
14	9	4	4	5	4	4	4	4	4	4
	12	6	6	6	6	6	6	6	6	6
15	9	2	2	2	3	1	2	2	2	2
	12	12	12	12	12	12	12	12	12	12
16	9	8	8	8	8	8	8	8	8	8
	12	8	8	8	8	8	8	8	8	8
17	9	3	1	3	3	3	4	3	4	3
	12	11	11	11	9	11	11	11	11	10
18	9	6	6	6	6	6	6	6	6	6
	12	4	5	4	4	4	4	4	4	4
19	9	2	2	2	4	2	2	2	2	2
	12	9	9	8	9	9	9	9	9	9
20	9	5	5	5	5	5	5	5	5	5
	12	10	10	10	10	10	10	10	10	10

표 3은 RP의 개수를 9와 12로 정해 놓고 K 값을 9, 10, 11, 12로 달리하여 실제 RP의 위치를 KNN과 제안한 알고리즘을 이용하여 추정하였다. 여기서 제안한 알고리즘을 이용한 반복 작업을 시작하기 전 초기값은 KNN 방법을 이용하여 나온 값을 사용하였다. 표 4는 20개의 실험 위치 중 다양한 경우에 따른 오버헤드와 알고리즘의 성공률을 나타낸 것이다. KNN 기법의 경우에는 K 가 9인 경우에서 10, 11, 12로 갈수록 약간씩 더 높은 추정 성공률을 보였으나 이것은 일반적인 경우로 판단하기보다는 미리 셋팅된 RP의 개수가 9 또는 12인 경우에 따라 다른 결과를 출력한 것으로 보인다. 여기서 K 가 증가함에 따라 발생하는 오버헤드는 평균 2.5%의 성공률이다. 한편 RP의 개

수가 9에서 12로 증가함에 따라 성공률은 평균 1.63% 감소한 것으로 측정되었다. 주목할 것은 K값이 커질수록 제안한 알고리즘으로 인한 위치 추정 성공률이 조금씩 더 향상되는데 이는 KNN 알고리즘 수행 후 나온 출력값을 제안한 알고리즘의 반복 작업 수행 시 초기 시작점으로 사용하면서 오차함수 상의 최적값을 찾아내는데 도움이 된 것으로 보인다. 각각의 출력값 일치를 평균을 보면 KNN과 제안한 알고리즘이, RP가 9와 12일 때 각각 82.5%, 81.25% 그리고 91.25%, 92.5%를 보이고 있다.

Table 4. Accuracy of the Location Estimation With respect to the Various K and RP.

	# of RP	Accuracy (KNN)	Accuracy (Proposed)
K=9	RP=9	80%	85%
	RP=12	80%	90%
K=10	RP=9	75%	90%
	RP=12	80%	90%
K=11	RP=9	85%	95%
	RP=12	80%	95%
K=12	RP=9	90%	95%
	RP=12	85%	95%
Avg.	RP=9	82.5%	91.25%
	RP=12	81.25%	92.5%

이는 종합적으로 볼 때 KNN에 이어 본 연구를 통해 제안된 알고리즘을 수행하는 방법이 매우 양호한 위치 추정 결과를 도출해내고 있음을 알 수 있다. 여기서 더욱 더 성공률을 높이기 위해서는 각 AP로부터 수신된 신호의 세기인 RSSI가 최대한 일정하고 정확하게 측정되어야 한다는 조건을 들 수 있다. 본 실험에서는 KNN 알고리즘의 조건으로 4개의 AP와 테스트 위치인 TP를 사용하고 또한 시행착오를 통해 얻어진 최종적인 K의 값으로 9부터 12까지의 다양한 값을 사용하였다. 그리고 오차 측정을 위해서는 현재의 실제 위치 $Q(x,y)$ 와 계산된 위치 $Q'(x',y')$ 또는 $Q'(x'',y'')$ 를 활용하였다. 더불어 KNN 기법과의 비교 분석을 위해 본 연구를 통해 제안된 방법을 적용하였는데 표 5에 경사하강법을 이용하여 제안한 알고리즘의 각 파라미터별 정의가 나타나 있다. 이러한 표 5의 조건을 이용하여 실험한 결과가 표 6에 나타난 바와 같이 도출되었다.

Table 5. Experimental Conditions used in the Proposed Algorithm

RP	$3 \times 3(9), 3 \times 4(12)$
Start	Output Coordinates after KNN
μ	0.001
Limits	$l > 0.0001$ and $m > 0.0001$

Table 6. Estimated Locations with Respect to the Number of Experiments (Err.:Error, Acc.:Accuracy)

#	Actual Location $Q(x,y)$	Estimated Location (KNN Algorithm)		Estimated Location (Proposed Methods)	
1	(1.25,1.12)	$Q'(x',y')$	(1.36,0.99)	$Q'(x'',y'')$	(1.3,1.15)
		Err.	0.17	Err.	0.06
		Acc.	83%	Acc.	94%
2	(2.25,1.25)	$Q'(x',y')$	(2.13,1.13)	$Q'(x'',y'')$	(2.22,1.21)
		Err.	0.17	Err.	0.05
		Acc.	83%	Acc.	95%
3	(1.75,1.75)	$Q'(x',y')$	(1.88,1.63)	$Q'(x'',y'')$	(1.77,1.8)
		Err.	0.18	Err.	0.05
		Acc.	82%	Acc.	95%
4	(2.12,0.87)	$Q'(x',y')$	(1.99,0.74)	$Q'(x'',y'')$	(2.14,0.89)
		Err.	0.18	Err.	0.03
		Acc.	82%	Acc.	97%
5	(1.25,3.25)	$Q'(x',y')$	(1.13,3.12)	$Q'(x'',y'')$	(1.22,3.27)
		Err.	0.18	Err.	0.04
		Acc.	82%	Acc.	96%
6	(1.37,2.25)	$Q'(x',y')$	(1.48,2.13)	$Q'(x'',y'')$	(1.42,2.28)
		Err.	0.16	Err.	0.06
		Acc.	84%	Acc.	94%
7	(2.75,2.75)	$Q'(x',y')$	(2.88,2.88)	$Q'(x'',y'')$	(2.77,2.71)
		Err.	0.18	Err.	0.04
		Acc.	82%	Acc.	96%
8	(2.75,3.25)	$Q'(x',y')$	(2.63,3.13)	$Q'(x'',y'')$	(2.79,3.3)
		Err.	0.17	Err.	0.05
		Acc.	83%	Acc.	95%
9	(1.12,3.12)	$Q'(x',y')$	(1.23,2.99)	$Q'(x'',y'')$	(1.08,3.14)
		Err.	0.17	Err.	0.04
		Acc.	83%	Acc.	96%
10	(1.62,3.5)	$Q'(x',y')$	(1.49,3.61)	$Q'(x'',y'')$	(1.59,3.47)
		Err.	0.17	Err.	0.04
		Acc.	83%	Acc.	96%
11	(2.12,4.12)	$Q'(x',y')$	(2.25,4.23)	$Q'(x'',y'')$	(2.14,4.17)
		Err.	0.17	Err.	0.05
		Acc.	83%	Acc.	95%
12	(2.37,3.25)	$Q'(x',y')$	(2.48,3.13)	$Q'(x'',y'')$	(2.4,3.28)
		Err.	0.16	Err.	0.04
		Acc.	84%	Acc.	96%
13	(2.12,2.87)	$Q'(x',y')$	(1.99,2.74)	$Q'(x'',y'')$	(2.17,2.83)
		Err.	0.18	Err.	0.06
		Acc.	82%	Acc.	94%
14	(1.75,2.75)	$Q'(x',y')$	(1.63,2.63)	$Q'(x'',y'')$	(1.77,2.72)
		Err.	0.17	Err.	0.04
		Acc.	83%	Acc.	96%
15	(1.12,1.75)	$Q'(x',y')$	(0.99,1.86)	$Q'(x'',y'')$	(1.14,1.8)
		Err.	0.17	Err.	0.05
		Acc.	83%	Acc.	95%
16	(2.12,2.12)	$Q'(x',y')$	(2.25,2.25)	$Q'(x'',y'')$	(2.09,2.08)
		Err.	0.18	Err.	0.05
		Acc.	82%	Acc.	95%
17	(2.62,2.25)	$Q'(x',y')$	(2.73,2.13)	$Q'(x'',y'')$	(2.58,2.27)
		Err.	0.16	Err.	0.04
		Acc.	84%	Acc.	96%
18	(1.75,3.87)	$Q'(x',y')$	(1.62,4.0)	$Q'(x'',y'')$	(1.8,3.9)
		Err.	0.18	Err.	0.06
		Acc.	82%	Acc.	94%
19	(2.12,3.75)	$Q'(x',y')$	(1.99,3.86)	$Q'(x'',y'')$	(2.15,3.8)
		Err.	0.17	Err.	0.06
		Acc.	83%	Acc.	94%
20	(2.62,3.75)	$Q'(x',y')$	(2.75,3.63)	$Q'(x'',y'')$	(2.67,3.72)
		Err.	0.18	Err.	0.06
		Acc.	82%	Acc.	94%
Average		Err.	0.1725	Err.	0.0485
		Acc.	82.75%	Acc.	95.15%

표 6은 현재 위치와 K 가 9이고, RP는 3×4 인 12개, TP는 50개일 때 KNN 기법을 이용하여 나온 좌표의 실험결과와 제안한 방법을 이용하여 경사하강법으로 위치의 좌표 값을 추정 한 결과를 20회 실험을 통하여 비교 분석해 놓은 결과이다. 좌표의 값은 상대적인 위치로, x와 y좌표 모두 좌표 1값 당 약 4m 정도의 실제 거리 차가 존재한다. KNN 방법을 통하여 추정된 결과 위치는 $Q'(x', y')$ 으로 표기 되었고 그렇게 추정된 값에 이어서 제안된 알고리즘을 추가로 적용한 최종 결과는 $Q''(x'', y'')$ 이다. 본 연구에서는 각각의 x, y좌표의 시작 위치를 0에 가까운 숫자인 (0.1, 0.1)에서 출발하도록 하였다. 또한 최종 결과 위치 $Q''(x'', y'')$ 를 구할 때 참조 위치의 수를 각각 다르게 하여 비교 분석해 보았다. μ 는 작으면 작을수록 더욱 더 세밀한 결과를 보이지만 수행 시간이 길어진다는 단점이 있기 때문에 적절히 상황을 고려하여 0.001로 조건을 주었다. μ 값의 변화에 따른 오버헤드는 10^{-1} 의 단위로 줄어들 때 위치 추정 시간이 대략 4000ms 증가하는 것으로 측정되었다. 이렇게 실험을 진행하여 KNN 알고리즘을 통해 나온 추정 위치 $Q'(x', y')$ 과 연이어 제안된 알고리즘을 적용시켜 나온 추정 위치 $Q''(x'', y'')$ 의 오차를 정확도와 함께 표 6에 보이고 있는데 여기서 오차는 현재의 실제 위치 $Q(x, y)$ 와 계산을 통해 결정된 KNN의 결과 위치 $Q'(x', y')$ 와 제안된 기법을 적용한 결과 위치 $Q''(x'', y'')$ 간의 거리의 차를 나타낸다. 따라서 표 6에 보이는 오차는 각각의 위치 좌표 간의 차이를 계산한 값으로, 작은 값일수록 짧은 거리, 즉 더 적은 오차를 나타내는 것이다. 여기서 오차는 수 식 (20)에 나타난 바와 같이 두 점 간의 거리 측정 방식의 가장 대표적인 방법인 Euclidean Distance 계산법을 사용하였다.

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (20)$$

그리고 정확도는 각각의 알고리즘을 적용하였을 때 현재의 실제 위치와 추정된 위치 간의 정확도를 측정하는 항목으로서 두 위치 간의 거리를 기반으로 표현하였는데 오차로 나타낸 거리의 차가 0일 때 100%, 1일 때 0%로 정확도의 기준을 선정하였고 전체 범위는 표 7에 나타나 있다.

Table 7. Precision Setting Range of the Estimated Location Results using the KNN and Proposed Methods

Error	Accuracy	Error	Accuracy
0	100%	0.51~0.6	49~40%
0.01~0.1	99~90%	0.61~0.7	39~30%
0.11~0.2	89~80%	0.71~0.8	29~20%
0.21~0.3	79~70%	0.81~0.9	19~10%
0.31~0.4	69~60%	0.91~1.0	9~0%
0.41~0.5	59~50%		

위치 측정 실험에서 최종적으로는 정확한 좌표를 얻어내는 것이 목표이기 때문에 본 연구에서는 참조 위치인 RP를 두어 해당 영역을 KNN을 이용하여 실제 위치로 추정하는 중간 단계

를 두었다. 이렇게 하여 얻어진 좌표 값을 이어서 경사하강법을 이용하는 제안된 알고리즘에 적용하여 최종 위치 좌표 $Q''(x'', y'')$ 을 찾아낸다. 표 6을 보면 RP 추정을 위한 KNN 이용 결과, 평균 정확도는 82.75%이고 연이어 이 결과를 이용하여 최종 추정 위치 결과 산출을 위한 제안된 기법 적용 결과 평균 95.15%의 정확도를 보였다.

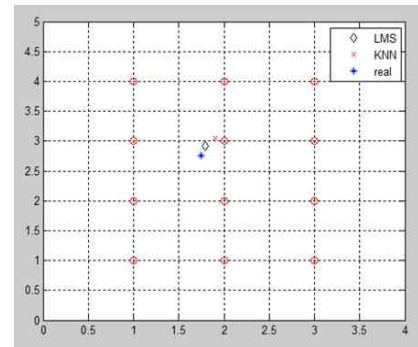


Fig. 13. Results from the KNN and the Proposed Algorithm (RP=Red Circles)

이는 KNN 기법에 이어 경사하강법을 이용하여 좌표 추정에 본 연구를 통해 제안한 알고리즘을 적용한 결과가 매우 양호함을 나타낸다고 볼 수 있다. 그림 13은 한 가지 특정 샘플에 대하여 현재의 실제 위치와 KNN 적용 후의 위치, 그리고 그 결과를 이용하여 제안된 알고리즘까지 적용하여 측정한 위치를 좌표 영역 안에 도형으로 나타낸 것이다. 위에 제시된 두 가지 방법을 통하여 추정된 좌표 (KNN:(1.89,3.03), 제안된 기법:(1.76,2.95)) 모두 좌표 (2, 3)에 위치한 최근접의 RP 영역 안에 존재하여 실험결과가 적절히 도출되었음을 알 수 있다. 그림 13을 보면 실제 위치, KNN 방법을 이용하여 추정한 위치, 제안된 방법을 이용하여 추정한 위치 등 세 가지 경우의 위치 좌표가 나타나 있다. 본 연구에서 제안한 알고리즘을 통해 얻은 결과는 KNN의 출력 좌표인 (1.89, 3.03)에서 추정 반복 작업을 시작하였는데 최적값을 찾는 반복 연산을 수행하면 보정 작업이 이루어져 조금 더 정확한 좌표를 도출해낸다. 따라서 KNN 측정치가 적절하고 정확할수록 제안된 알고리즘의 수행 결과 또한 뛰어나다고 할 수 있다. 두 알고리즘을 연이어 사용하면 무선 랜 환경에서 AP의 수신 신호세기 측정을 통해 현재 사용자의 위치를 비교적 정확하게 알아낼 수 있고 IoT와 M2M 시대에 걸맞은 스마트 홈 구현을 위해 응용이 가능할 것으로 사료된다. 다음 절에는 임베디드 장비를 이용한 실험을 통해 스마트 홈 시스템을 가상하여 직접 구현해 보았다.

2. Implementation of a Smart Home System

지금까지의 실험에서는 Fingerprint를 이용한 KNN 알고리즘과 Gradient Descent Algorithm을 활용하여 제안된 알고리즘을 이용하여 RSSI 신호세기에 따른 이동 노드의 위치를 추정하였다. 본 실험에서는 제안된 기법을 이용하여 곳곳에 네트워크

크 통신이 가능한 고정기기들을 설치하여 그림 14에 나타난 바와 같이 소규모의 가전 시스템이 동작하는 가상의 스마트 홈을 구축하였다. 그림 14에 나타난 공간의 각 모서리에 위치 측정을 위한 AP (AP1-AP4)들을 설치하였고, 각각 4곳의 위치에 고정기기 (D1-D4)들을 설치하였다.

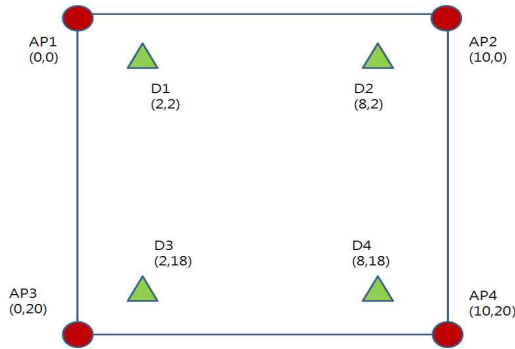


Fig. 14. Installed Structure and Coordinates of Access Points (AP1-AP4) and Devices (D1-D4)

그리고 고정 기기에 무선 통신으로 연결되는 이동노드가 움직일 때 마다 통신을 하여 이동 노드와 가장 가까운 위치에 있는 장치가 자신의 특성에 맞게 자동으로 적응적으로 동작하는 시스템으로 구성되었다. 각 고정기기들은 그림 15의 (a)에 보이는 바와 같이 네트워크 통신이 가능한 임베디드 장비인 Raspberry PI를 이용하였고 각각 4개의 장치에 가정에서 흔히 이용하는 가전제품들로 가정하고 연결하여 실험을 진행하였다. D1에는 그림 15의 (b)에 나온 것처럼 일반 가정의 현관에 장착되어 있는 문 잠금 장치를 가정하여 Digital Door-lock을 연결하였고 D2와 D4에는 일반 가정 내의 실내등을 가정하고 연결하여 조명을 제어하도록 했다. 마지막으로 D3에는 미니 선풍기를 연결하여 자동으로 동작 되도록 하였다. D1은 자동 문 잠금 장치로서 이동 노드에 해당하는 사람이 D1 근처에 오면 자동으로 문 잠금과 해제 동작을 하도록 했고 D2와 D4 역시 근거리에서 이동 노드의 움직임이 감지되면 조명의 점등과 소등 동작을 하도록 하였다. D3는 근거리에서 이동 노드에 해당하는 움직임이 감지되면, 즉 위치 추정 알고리즘을 통해 RSSI 세기에 의한 측정이 제대로 이루어져 이동 노드에 해당하는 사람의 위치가 파악되면, 가전기기에 해당하는 선풍기가 동작하도록 실험 장치들을 배치하였다.

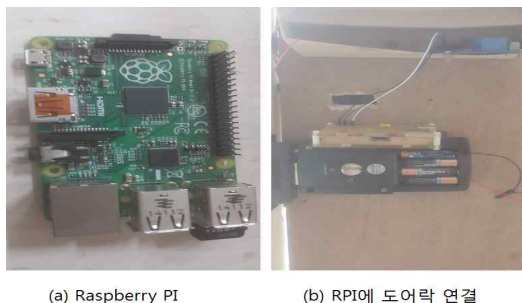


Fig. 15. Embedded Board (Raspberry PI) and the Digital Door-Lock (D1) used as Fixed Device

좌표 내에서 임의로 움직이는 이동 노드로는 실험에 참가한 사람이 소지한, 안드로이드 플랫폼을 기반으로 하는 스마트 폰을 사용하였고 이 스마트 폰과 고정기기인 임베디드 장치와 (Raspberry PI, D1-D4)와 UDP 통신을 하도록 했다. 실험 진행을 위해 스마트 폰을 소지하고 사용자가 이동했을 때 사용자와 각각 장치들 (D1-D4)간의 거리를 계산하여 이동 노드인 스마트 폰 화면에 나타내고 이들 장치들 중 가장 가까운 위치에 있는 장비에만 'On' 명령을 내려 동작하도록 하였다. 사용자가 그림 14에 보이는 가상의 스마트 홈 공간에 들어섰을 때 AP로부터 스마트 폰이 받는 신호세기인 RSSI의 갱신 시간은 약 3초이며 실험 진행을 위해 임의의 위치에서 약 90초 동안 각각 100번의 측정을 하고 위치 이동을 하면서 실험을 진행하였다. 그림 16과 Table 8에 이동 노드에 해당하는 스마트 폰을 가진 사용자의 실제 위치가 (2,16)인 경우 KNN과 제안된 기법을 이용하여 위치 측정 실험을 하고 현재 위치를 측정한 예가 나타나 있다. 그 후 측정된 현재 위치와 고정된 기기들 간의 거리를 측정하여 가장 거리가 짧은 위치에 있는 기기를 찾는다. 그림 16의 경우에는 이동 노드인 스마트 폰의 위치가 D3와 가장 가깝기 때문에 그림 14에 나온 (2,18) 좌표 값을 가진 장치 D3가 사용자와 가장 가까운 거리에 위치하고 있다 할 수 있다. 여기서 스마트 폰의 위치와 D3간의 실제 거리는 약 2m정도인데 100회 측정 후 관측된 D3와 이동 노드의 평균 거리는 약 1.69m였다. 이는 적절하게 두 장치간의 거리 측정을 한 경우와 그렇지 않은 경우들의 평균 거리인데 결과적으로는 실제거리보다 더 가깝게 나와 매우 신뢰할 만한 최근접 거리 측정 방법을 사용한 것이라 할 수 있다.

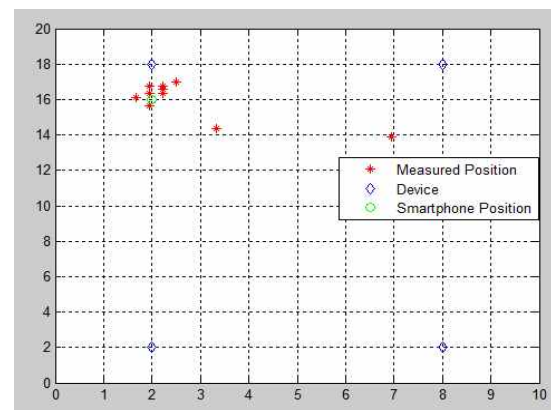


Fig. 16. Connection Result between the Smart Phone and the Nearest Device (D3) after the Location Estimation

Table 8. Experiment Results of a Location Estimation

Moving Node Location (SP)	(2,16)
Closest Device Location (DP)	(2,18)
Measured average distance (SP-DP)	1.69m
DP Connection Success Rate	95.0%(95/100)

또한 D3에 해당하는 실내등 조명에 'On' 명령어를 성공적으로 전달한 경우가 100번 중에 95번으로 성공률은 약 95%였다. 이렇듯 다양한 위치에서의 이동 노드와 장치들 (D1-D4)간의 거리를 측정하여 최근접 장치들의 동작을 자동으로 하게 하는 가상 스마트 홈 시스템의 작동 결과가 표 9에 나타나 있다.

Table 9. Average Distance and Connection Success Rate after the Estimation of the Distance between the Moving Node and Fixed Device in Virtual Smart Home System (N: No obstacles, M: Metal obstacles)

		D1	D2	D3	D4
Device Location		(2,2)	(8,2)	(2,18)	(8,18)
Moving Terminal Location		(4,2)	(8,6)	(2,15)	(8,13)
Actual Distance		2	4	3	5
Average Distance		1.85	4.24	3.11	5.16
Connection Success Rate	N	96.0% (96/100)	91.0% (91/100)	94.0% (94/100)	95.0% (95/100)
	M	93.0% (93/100)	85.0% (85/100)	87.0% (87/100)	86.0% (86/100)

네 개의 고정 장치와 하나의 이동 노드인 스마트 단말을 이용한 가상의 스마트 홈에서의 실험 결과 각 장치에 대한 이동 단말의 추정 위치는 실제거리와 평균거리의 차이를 구했을 때 평균 0.165 미터의 오차를 보였다. 또한 이동 스마트 단말과 최근접의 위치에 있는 고정 임베디드 장치들 간의 접속 후 동작 성공률의 평균 또한 100회씩 측정하여 분석했을 때 전체 횟수의 94.0%에 해당하는 94번 이었다. 또한 현실 속에서 우리는 다양한 금속 물질이 공존하는 환경에서 생활하므로 이러한 상황을 가정하여 TV, 냉장고, 컴퓨터 등의 메탈 물질이 있는 공간에서 실험을 진행하였고 이 경우, 동작 성공률은 전체 횟수의 87.75%에 해당하는 87.75번 이었다. 이러한 결과를 분석해보면 본 연구를 통해 제안한 기법의 위치 추정 성능이 매우 양호함을 알 수 있다. 본 실험에서는 무선 랜 환경에서 각각의 AP로부터의 신호세기 측정을 통하여 이동 장치의 현재 위치를 추정하고, 고정된 장치의 위치와 각각의 거리를 계산한 후 최근접 장치를 사용자의 기호에 맞게 동작 시키는 가상 스마트 홈 시스템을 구현하고 그 성능을 분석해 보았다. 본 실험 결과는 본격적인 스마트 홈 시스템 구축이 현실적으로 상용화 가능한 시점이 되면 집안 가전에 해당하는 에어컨, 냉장고, 가습기, 냉온수기, 공기청정기, 청소기 등의 가전 장비에 연결되어 IoT 분야에서 응용이 가능하리라 사료된다.

V. Conclusions

현재 그리고 앞으로 다가올 미래의 ICT에서의 중점 분야 중 하나는 M2M과 IoT에 관련된 기술들이다. 그 중에서도 사람의 생활과 가장 밀접한 부분인 스마트 홈 기술은 현재에도 다양한

영역에서 연구가 진행되고 있는 중이다. 스마트한 가전의 자율 동작을 위해 실외와 같은 넓은 지역이 아닌 좁은 실내에서의 위치 측정 또한 다양한 방법을 통하여 정확도를 높이고자 하는 시도가 이루어지고 있다. 본 논문에서는 스마트 홈 구축을 위한 실내 무선 랜 환경에서의 위치 측정에 관한 연구를 진행하였다. Fingerprint 방식을 이용하여 참조 위치인 RP에서 RSSI 신호 세기를 획득하고 그 후 특정의 한 지점에서의 신호세기와 기존의 저장된 값을 이용하여 현재의 위치를 추정하였다. 이를 위해 패턴인식 및 추론 분야에서 활용되는 KNN 알고리즘을 이용하여 실험을 통한 경험적인 파라미터로 다양한 K 값을 사용하여 위치를 도출해냈고 이후에 기계학습 분야에서 응용이 가능한, 제안된 알고리즘으로 오차를 보정하면서 실험을 진행하였다. 본 연구를 통해 제안한 알고리즘에서는 경사하강법을 이용하여 추정하고자 하는 이동 노드의 x와 y의 좌표 파라미터에 대하여 각각 편미분을 수행함으로써 오차 함수를 계산하고 최저의 오차값을 주는, 다시 말해서 최적의 x와 y의 좌표를 찾아내어 이동하는 스마트 단말의 위치를 추정하였다. 이 때 각 지점에서의 신호세기는 수시로 변하지만 명확한 값을 가지면 가질수록 위치 추정의 정확률은 높아진다. 본 논문에서는 신호세기에 해당하는 RSSI를 측정하여 이동 단말과 고정 장치 간의 거리를 계산, 위치 추정을 하였고 거리 계산 결과가 양호할수록 최종 실험 결과의 정확성과 성공률은 보장된다고 할 수 있다. 또한 각각의 상황에 맞게 알고리즘이 적용되어야 하는데 이 때 본 연구를 통해 제안된 알고리즘을 수행할 시 적절한 K값 지정과 위치 추정을 위한 x와 y의 좌표의 오차 함수 편미분 계산 과정 또한 매우 중요하다. 본 연구를 통하여 사용자 편의를 위해 최근접 거리에 위치한 가전 장치의 자동 동작으로 가상 스마트 홈을 구현해 보았고 가장 기초가 되는 기기들 간의 통신을 이용한 서비스 환경을 구축하고 연구를 진행하였다. 앞으로 이 환경과 조건을 좀 더 응용하고 확장하여 또 다른 통신 환경에서 다양한 장치들이 연결되어 자율적으로 동작하는 스마트 홈이 포함된 M2H 그리고 IoT 관련 연구를 지속적으로 수행할 계획이다.

REFERENCES

- [1] J. Kim, et al., "M2M service platforms: survey, issues, and enabling technologies," IEEE Communications Surveys & Tutorials, Vol. 16, No. 1, pp. 61-76., October 2014.
- [2] M. Chen, J. Wan, and F. Li, "Machine-to-Machine Communications," KSII Transactions on Internet and Information Systems, Vol. 6, No. 2, pp. 480-497., February 2012.
- [3] S. Jung, B. Moon, and D. Han, "Unsupervised Learning for Crowdsourced Indoor Localization in Wireless

- Networks," *IEEE Transactions on Mobile Computing*, Vol. 15, No. 11, pp. 2892-2906., December 2016.
- [4] J. Suh, S. You, S. Choi, and S. Oh, "Vision-Based Coordinated Localization for Mobile Sensor Networks," *IEEE Transactions on Automation Science and Engineering*, Vol. 13, No. 2, pp. 611-620., November 2016.
- [5] J. Li, D. Liu, and B. Yang, "Smart home research," *Proceedings of the Third Conference on Machine Learning and Cybernetics SHANGHAI*, August 2004.
- [6] W. Haidong, J. Saboune, and A. Saddik, "Control your smart home with an autonomously mobile smartphone," *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, July 2013.
- [7] A. E. Ruiz, E. R. Cruz, M. J. T. Urrea, E. C. R. Lbarra, J. R. Lbarra, and J. C. Gonzalez, "Performance comparison between simulated and real case scenario of RSSI-Based localization algorithms on a WSN," *IEEE Latin America Transactions*, Vol. 14, No. 1, pp. 115-121., March 2016.
- [8] Y. Chen, D. Deng, and C. Teng, "Range-Based Localization Algorithm for Next Generation Wireless Networks Using Radical Centers," *IEEE Access*, Vol. 4, pp. 2139-2153., April 2016.
- [9] K. Lee, C. Chae, T. Sung, J. Kang, "Cognitive beamforming based smart metering for coexistence with wireless local area networks," *Journal of Communication and Networks*, Vol. 14, No. 6, pp. 619-628., January 2013.
- [10] Z. Liu, and H. Wei, "The design of smart home system based on Wi-Fi," *IEEE International Conference on Computational Problem-Solving (ICCP)*, October 2012.
- [11] S. Tomic, M. Beko, and R. Dinis, "RSS-Based Localization in Wireless Sensor Networks Using Convex Relaxation: Noncooperative and Cooperative Schemes," *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 5, pp. 2037-2050., July 2016.
- [12] Y. Wu, H. Wang, and X. Zheng, "WSN Localization Using RSS in Three-Dimensional Space—A Geometric Method With Closed-Form Solution," *IEEE Sensors Journal*, Vol. 16, No. 11, pp. 4397-4404., March 2016.
- [13] M. Umair, V. Kopparapu, and Y. Dongkai, "An enhanced K-Nearest Neighbor algorithm for indoor positioning systems in a WLAN," *IEEE Computing, Communications and IT Applications Conference (ComComAp)*, October 2014.
- [14] G. Zou, et al., "An indoor positioning algorithm using joint information entropy based on WLAN fingerprint," *IEEE Conference on Computing, Communication and Networking Technologies*, July 2014.
- [15] S. Lee, "A Comparison of the Localization Technology in Wireless Sensor Network," *Inha University Press*, February 2008.
- [16] R. Sanz, P. Corral, and A. Lima, "Adaptive beamforming techniques for OFDM based WLAN systems: a comparison between RLS and LMS," *Mobile Future, and the Symposium on Trends in Communications.*, October 2004.
- [17] B. Mohr, W. Li, and S. Heinen, "Analysis of digital predistortion architectures for direct digital-to-RF transmitter systems," *IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, August 2012.
- [18] J. Dholakia, V. Jain, and B. Myers, "Adaptive equalization for 100 Mbps OWSS wireless LANs," *IEEE Global Telecommunications Conference*, Vol. 6., November 2001.
- [19] Q. Yang, and J. Sun, "Study on location system of underwater robot based on LMS adaptive algorithm," *IEEE-WCICA 7th World Congress on Intelligent Control and Automation*, June 2008.
- [20] S. Yi, and L. Haijun, "An Adaptive Localization Method for Autonomous Digging Robot," *IEEE Control Conference*, July 2007.

Authors



Daewon Kim received a M.S. (1996) from the University of Southern California, LA, California, USA, and a Ph. D. (2002) in Electrical and Computer Engineering from Iowa State University, Ames, Iowa, USA. Daewon Kim worked as a senior researcher at Samsung Electronics Co. Ltd., Suwon, South Korea (2002-2004). He is currently a professor in Applied Computer Engineering department at Dankook University, South Korea. His research interests include signal processing, mobile applications, and nondestructive evaluation.