

WLAN 기반 Extended Kalma Filter 추적의 정확성 제고 방법

임재걸*, 정승환*, 김윤호*

*동국대학교 컴퓨터멀티미디어학과

e-mail: {yim, honourj, dbsh8577}@dongguk.ac.kr

Techniques to Improve Accuracy of WLAN-Based Extended Kalman Filter Tracking

Jaegel Yim*, Seunghwan Jeong*, Yunho Kim*

*Dept of Computer and Multimedia, Dongguk University

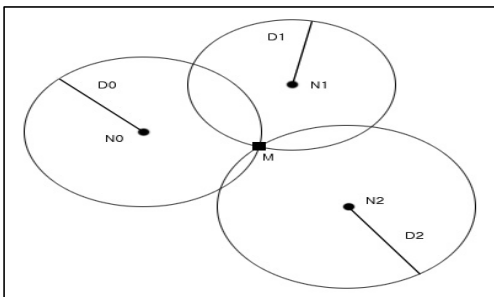
요 약

위치기반서비스에서 사용자의 정확한 위치가 요구되면서 옥내 위치기반서비스를 위한 무선 LAN기반 옥내 측위와 추적에 대한 연구가 활발히 진행되고 있다. 본 연구에서는 AP 신호의 세기를 바탕으로 거리와 신호세기의 관계를 통해 사용자의 위치에서 AP들까지의 거리를 구한 후, Trilateration으로 위치를 추적하는 방법과 EKF 방법을 비교한다. EKF를 이용할 때 필터링에 사용되는 파라메타들의 값에 따라 달라지는 실험 결과들을 비교 분석하여 적절한 파라메타 값을 찾고, 나아가서 파라메타 값을 적절히 적용함으로써 경로의 정확성을 제고하는 방법을 제시한다. 제안하는 방법의 효율성을 실험적으로 증명하기 위한 실험 결과와 분석 내용도 제시한다.

1. 서론

본 연구에서는 무선로컬네트워크(WLAN) 환경이 구축된 옥내에서 사용자의 이동 경로를 추적하는 Extended Kalman Filter(EKF)[1] 방법을 살펴본다. 이동 경로는 일정 시간이 경과한 때 시점마다 사용자의 위치를 측정하여 구할 수 있는데, 이렇게 구한 경로는 측위 방법의 오류 때문에 실제 경로와 차이가 많이 난다. 본 연구에서는 EKF를 이용하여 이 차이를 줄이는 방법을 제시한다.

WLAN 환경에서 AP들의 신호세기(RSS)를 측정할 후 거리와 신호세기간의 관계식(propagation model이라 함)[2]을 이용하여 사용자의 위치로부터 AP들까지의 거리들을 계산한다. 매 시점마다 이렇게 구한 거리들을 가지고 Trilateration 방법[3]으로 사용자의 위치를 판단할 수 있다. 그러나 본 연구에서는 Trilateration 방법 대신 EKF 방법을 이용할 경우 더욱 정확한 경로를 작성할 수 있다는 것을 보이고, EKF를 이용할 때 필터링에 사용되는 파라메타들의 값에 따라 달라지는 실험 결과들을 비교 분석하여 적절한 파라메타 값을 찾고, 나아가서 파라메타 값을 적절히 적용함으로써 경로의 정확성을 제고하는 방법을 제시한다.



(그림 1) Trilateration 방법을 나타내는 도형

2. 기존의 연구

본 절에서는 기존의 측위 방법 중 Trilateration 방법과 추적 방법 중 EKF 방법을 소개한다. 사용자의 위치를 Mobile device의 위치라 하고 AP들을 Neighbor device라 한다.

2.1 Trilateration 방법

Trilateration 방법이 적용되려면 그림 1에 보이는 바와 같이 Mobile device에서 최소 3개의 Neighbor device까지의 거리와 위치를 알아야 한다. 그림에서 N_0, N_1, N_2 는 알려진 Neighbor device인 AP들의 위치이며, D_0, D_1, D_2 는 Mobile device인 사용자(M)의 위치로부터 AP들까지의 거리이다. AP를 중심으로, 이들 AP와 사용자간의 거리를 반지름으로 하는 원 또는 구를 형성할 때 생성되는 교차점이 바로 사용자의 위치가 된다. 일반적으로 M 과 N_i 의 위치를 각각 $(x, y, z), (X_i, Y_i, Z_i)$ 좌표로, 그리고 M 과 N_i 간의 거리를 D_i 로 나타내면 좌표와 거리와의 관계식은 다음과 같다.

$$(x - X_i)^2 + (y - Y_i)^2 + (z - Z_i)^2 = D_i^2$$

$$(i = 0, 1, 2, \dots, m-1)$$

3차원일 경우 최소한 4개의 고정 노드가 필요하며, 위의 식에서 제곱 항을 소거하면, 3개의 변수를 갖는 선형 방정식을 구할 수 있고 이를 Matrix 형태로 표현하면 $\vec{Ax} = \vec{b}$ 가 된다. 여기에서 A, \vec{x}, \vec{b} 는 다음과 같다.

$$A = \begin{bmatrix} 2(X_1 - X_0) & 2(Y_1 - Y_0) & 2(Z_1 - Z_0) \\ 2(X_2 - X_0) & 2(Y_2 - Y_0) & 2(Z_2 - Z_0) \\ 2(X_3 - X_0) & 2(Y_3 - Y_0) & 2(Z_3 - Z_0) \\ \dots & \dots & \dots \\ 2(X_{m-1} - X_0) & 2(Y_{m-1} - Y_0) & 2(Z_{m-1} - Z_0) \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} (X_1^2 - X_0^2) + (Y_1^2 + Y_0^2) + (Z_1^2 + Z_0^2) - (D_1^2 + D_0^2) \\ (X_2^2 - X_0^2) + (Y_2^2 + Y_0^2) + (Z_2^2 + Z_0^2) - (D_2^2 + D_0^2) \\ (X_3^2 - X_0^2) + (Y_3^2 + Y_0^2) + (Z_3^2 + Z_0^2) - (D_3^2 + D_0^2) \\ \dots \\ (X_{m-1}^2 - X_0^2) + (Y_{m-1}^2 + Y_0^2) + (Z_{m-1}^2 + Z_0^2) - (D_{m-1}^2 + D_0^2) \end{bmatrix}$$

위의 식을 풀기 위해서는 다음 식에서 δ 를 최소로 하는 \vec{x} 를 구한다.

$$\delta = (A\vec{x} - \vec{b})^T (A\vec{x} - \vec{b}), \quad \vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

\vec{x} 는 MMSE(Minimum Mean Square Error)방식에 의한 Matrix 해답으로 다음과 같이 구할 수 있다.

$$\vec{x} = (A^T A)^{-1} A^T \vec{b} \quad \text{--- (식1)}$$

2.2 Extended Kalman Filter

EKF 기술은 상태변수의 오차공분산 초기값(P_0^-), 프로세스 잡음강도(Q), 그리고 측정 잡음 공분산값(R)의 적절한 파라미터를 사용할 때 최적의 성능을 낼 수 있다. 모르는 위치의 Mobile device, $A = [x \ y \ z]^T$ 에서 알고 있는 위치의 Neighbor device, $(X^i, Y^i, Z^i), i = 1, \dots, m$ 에 대하여 측정된 거리는 다음 식으로 나타낼 수 있다.

$$r^i = \sqrt{(X^i - x)^2 + (Y^i - y)^2 + (Z^i - z)^2} + v^i$$

여기서 v^i 는 측정오차를 나타내며 평균 0, 분산 σ_v^2 인 백색 Gaussian 잡음(AWGN)으로 가정한다. $m(\geq 3)$ 개 Neighbor device에서 거리측정치를 구했다면 WLSQ (Weighted Least Square)기법으로 Node의 위치를 구할 수 있으며, $r^0 = (x_0 \ y_0 \ z_0)^T$ 을 기준으로 선형화하면 다음과 같고

$$\begin{bmatrix} r^1 - r_0^1 \\ \dots \\ r^m - r_0^m \end{bmatrix} = \begin{bmatrix} h_x^1 & h_y^1 & h_z^1 \\ \dots & \dots & \dots \\ h_x^m & h_y^m & h_z^m \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} + \begin{bmatrix} v^1 \\ \dots \\ v^m \end{bmatrix}$$

이를 다시 다음과 같은 간략한 식으로 표현할 수 있다.

$$\delta r = H \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} + v$$

여기서 $r_0^i = \sqrt{(X^i - x_0)^2 + (Y^i - y_0)^2 + (Z^i - z_0)^2}$ 는 알고 있는 Neighbor device의 위치와 기준점으로 계산한 거리이며,

$$(h_x^i = \frac{X^i - x_0}{r_0^i}, \quad h_y^i = \frac{Y^i - y_0}{r_0^i}, \quad h_z^i = \frac{Z^i - z_0}{r_0^i})$$

는 기준점에서

Neighbor device로의 시선(LOS: Line Of Sight) 벡터를 나타낸다. 시스템 모델에는 여러 가지가 있을 수 있는데 항법에서는 P(Position: 정지), V(Velocity: 등속), A(Acceleration: 등가속) Model을 사용하는 것이 일반적이다. 움직이는 경우가 아닌 정지 상태의 사용자의 위치를 측정한다면, 정지 모델을 이용하여 다음과 같은 식을 사용한다.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} + \begin{bmatrix} w_{xk} \\ w_{yk} \\ w_{zk} \end{bmatrix}$$

여기서 시스템 오차 $W_k \sim N(0, Q_k)$ 는 모델링 오차를 나타낸

것으로, 정지 상태의 사용자의 위치를 측정하는 경우에는 $Q_k = 0$ 으로 한다. 시스템 모델과 측정 모델을 일반화하면 $X_{k+1} = X_k + W_k, \quad Z_k = h(X_k) + V_k$ 이고, Kalman Filter를 적용하기 위하여 비선형 측정식을 선형화하면 $\delta X_{k+1} = \phi_k \delta X_k + W_k, \quad \delta Z_k = H_k \delta X_k + V_k$ 이다. 이때 구해진 값은 $\delta \hat{X}_k$ 이며 최종 위치는 $\hat{X}_{k+1} = X_0 + \delta \hat{X}_k$ 로 구하면 된다. 오차 $\delta \hat{X}_k$ 가 아닌 전체 \hat{X}_k 를 state로 두고 Extended Kalman Filter를 구성하면 최종 위치는 다음과 같은 식으로 구한다.

$$\hat{X}_{k+1} = \hat{X}_k + K_{k+1}(Z_{k+1} - \hat{Z}_{k+1}), \quad (\hat{Z}_{k+1} = h(\hat{X}_k))$$

표 1은 EKF의 시스템 모델과 측정 모델을 정리하여 최종 위치를 구하는 과정이 보이고 있다. 여기서 \hat{r}_k 는 신호세기로부터 거리로 계산된 값이며, 예측 상태 \hat{r}_{k+1}^- 는 사용자의 위치라고 예측한 좌표이다.

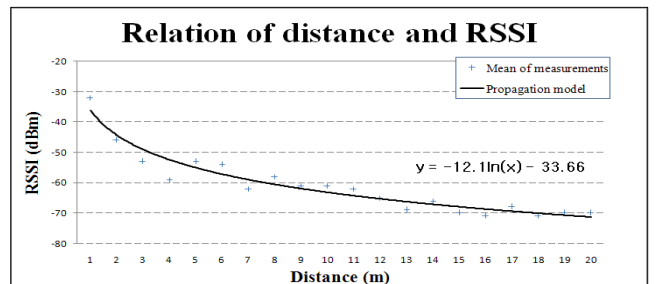
<표 1> EKF의 최종 위치를 구하는 과정

Linearized state model:
 $\delta \hat{x}_{k+1} = \Phi_k \delta \hat{x}_k + \hat{w}_k, \hat{w}_k \sim N(0, Q_k)$

Linearized measurement model:
 $\hat{r}_k = \hat{r}_0 + H_k \delta \hat{x}_k + \hat{v}_k, \hat{v}_k \sim N(0, R_k)$

- 1) Initial guess:
 $\hat{x}_0^- = E(x_0)$ and $P_0^- = \text{var}(\hat{x}_0)$
- 2) Linearizing:
 $\hat{r}_k = \hat{r}_0 + H_k \delta \hat{x}_k + \hat{v}_k,$

$$\hat{r}_0 = \begin{bmatrix} \sqrt{(X_1 - x_k^-)^2 + (Y_1 - y_k^-)^2 + (Z_1 - z_k^-)^2} \\ \dots \\ \sqrt{(X_N - x_k^-)^2 + (Y_N - y_k^-)^2 + (Z_N - z_k^-)^2} \end{bmatrix}$$
- 3) Kalman Gain:
 $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$
- 4) Measurement update:
 $\delta \hat{x}_k = \delta \hat{x}_k^- + K_k (\hat{r}_k - \hat{r}_0^-), \hat{x}_k = \hat{x}_k^- + \delta \hat{x}_k$
- 5) Update error covariance:
 $P_k = (I - K_k H_k) P_k^-$
- 6) State propagation:
 $\hat{x}_{k+1}^- = \Phi_k \hat{x}_k, \quad P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k$
- 7) Goto step2



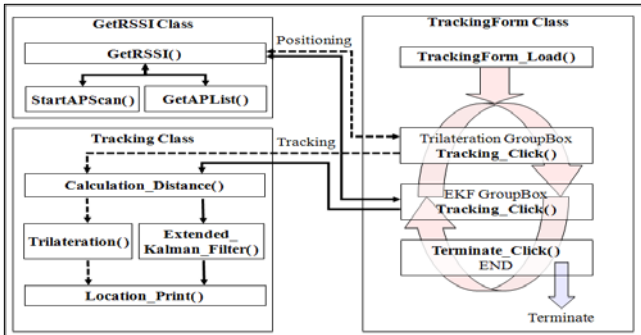
(그림 2) 거리와 신호세기의 관계

3. 사용자 추적 시스템

WLAN 환경에서 거리와 신호세기의 관계를 구하기 위해 AP로부터 1m 거리마다 RSS를 300회 측정한 평균을 구하여 그림 2와 같이 그래프로 나타냈다. X-축은 AP와 측정 장소 간의 거리이고, Y-축은 측정된 RSS의 평균값이다. 실선은 엑셀로 구한 거리와 신호세기의 관계식이며, 이 식을 이용

하여 사용자의 위치에서 AP들까지의 거리를 구할 수 있다.

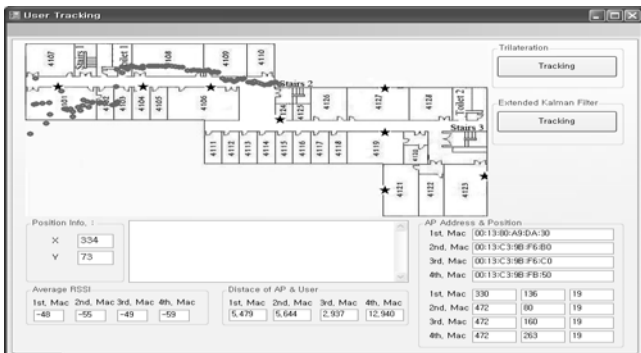
이 시스템은 Intel(R) PRO/Wireless 2200BG Network Connection 랜 카드가 장착된 Lab-Top 컴퓨터에서 Microsoft Visual Studio .NET 2005의 C#언어를 사용하여 구현한다. 본 절에 소개되는 추적 시스템에서는 행렬을 이용한 연산이 주로 사용되는데, 이 문제를 해결하기 위하여 행렬의 요소를 Decimal Type(C#)까지 계산할 수 있는 Matrix Class를 작성하였다. 이 Matrix Class는 행렬의 합, 곱 등의 연산과 전치행렬, 역행렬을 구할 수 있으며, 역행렬일 경우 여인자전개법의 공식을 이용하고 있다.



(그림 3) 사용자 추적 시스템의 이벤트 흐름도

3.1 Trilateration 방법 구현

Mobile device M의 좌표를 식 1에 적용하여 구하는 프로그램의 구조는 그림 3에 포함되어 있다. Trilateration GroupBox의 Tracking_Click() 이벤트가 발생되면 GetRSSI Class에서 신호세기를 측정한다. 그 다음 측정된 신호세기를 Positioning Class에서 Calculation_Distance() 함수를 통해 사용자 위치로부터 AP들까지의 거리를 구하고, Trilateration() 함수로 사용자의 위치를 판단하면, Location_Print() 함수가 위치를 출력한다.



(그림 4) 사용자 추적 시스템의 실행화면

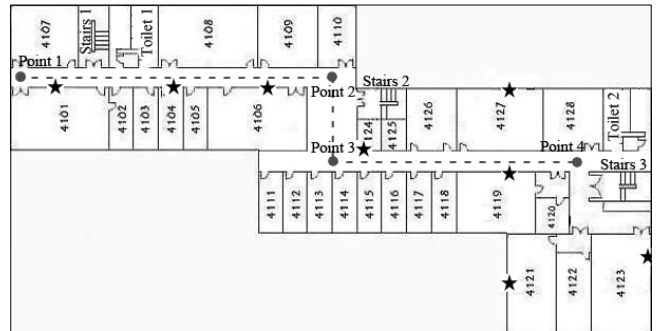
3.2 Extended Kalman Filter 방법 구현

2절에 소개된 사용자 추적을 위한 EKF의를 이용하여 사용자의 위치를 추적하는 프로그램의 구조가 그림 3에 포함되어 있다. EKF GroupBox의 Tracking_Click() 이벤트가 발생되면 GetRSSI Class에서 신호세기를 측정한다. 그 다음 측정된 신호세기를 Positioning Class에서 Calculation_Distance() 함수를 통해 사용자 위치로부터 AP들까지의 거리를 구하고, Extended_Kalman_Filter() 함수로 사용자의 위치를 판단하면, Location_Print() 함수가 위치를 출력한다.

그림 4는 사용자 추적 시스템을 실행한 화면으로 Trilateration GroupBox의 Tracking 버튼을 클릭할 경우엔 Trilateration 방법으로 사용자의 경로를 추적하며, EKF GroupBox의 Tracking 버튼을 클릭할 경우엔 EKF 방법으로 경로를 추적한다.

4. 실험

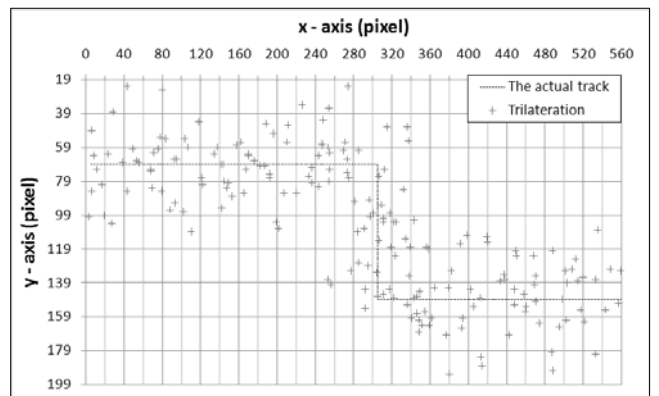
본 연구에서는 WLAN 환경의 IEEE 802.11b 규격의 2.4GHz RF의 RSS를 이용하여 그림 5에 보이는 자연과학관 4층(85.5m x 43.8m) 복도에서 사용자 추적 시스템을 실험하였다.



(그림 5) 자연과학관 4층

4.1 실험 환경 및 내용

실험에 사용된 AP들은 총 8개로 AP1부터 AP8까지 그림 2에 '★'로 표시되었다. 사용자의 이동 경로는 그림 2의 점선이며 좌측 상단의 Point 1을 시작으로 우측 중앙의 Point 4까지로 한다. EKF를 이용할 경우 사용자 추적 시스템의 초기 파라메타로 r_0^- 행렬의 모든 요소들을 0으로 하고, P_0^- 행렬의 대각선상의 요소들을 100으로 설정한다.



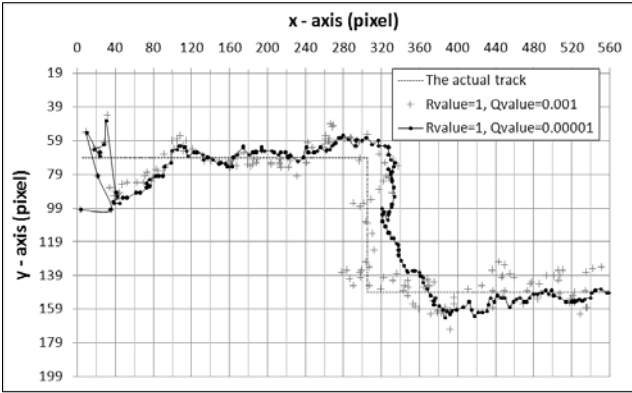
(그림 6) 실제 경로와 Trilateration의 결과

4.2 실험 결과

실제 경로를 걸어가면서 매초 3회씩 Trilateration 측위 실험을 100회 반복하여 Trilateration 측위의 평균을 구한 결과를 그림 6에 실제 경로(The actual track)와 함께 나타내었다. 이 경우 평균 오차는 3.45m이다. 그리고 상태 오차 $Q(6 \times 6 \text{ matrix})$ 와 측정치 오차 $R(4 \times 4 \text{ matrix})$ 의 대각선상의 원소의 값을 R_{value} 와 Q_{value} 라 하고, EKF의 파라메타를 표 2와 같이 설정하여 실험한 결과가 그림 7에 보이고 있다.

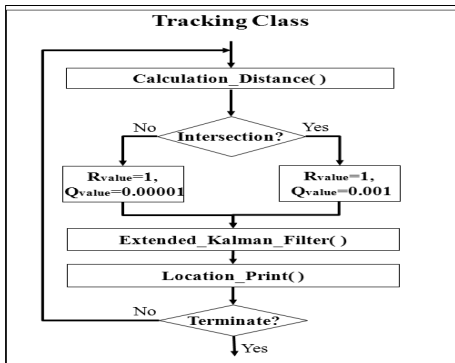
<표 2> 행렬 R과 Q의 R_{value}와 Q_{value} 설정

실험 index	R의 대각선상의 원소 R _{value}	Q의 대각선상의 원소 Q _{value}
실험 1	1	0.001
실험 2	1	0.00001



(그림 7) EKF의 표 2에 따른 실험 결과

그림 7은 EKF의 파라메타를 표 2와 같이 설정하여 실험한 결과이다. 실험 1과 실험 2에서 R_{value}가 동일한 상황에서 Q_{value}의 차이로 결과가 달라지는 것을 볼 수 있다. 실험 1의 경우 평균 오차는 1.81m로 경로가 분산되는 경향이 있으며, 실험 2의 경우 평균 오차는 2.33m로 사용자의 경로가 완만하게 추적되었다.



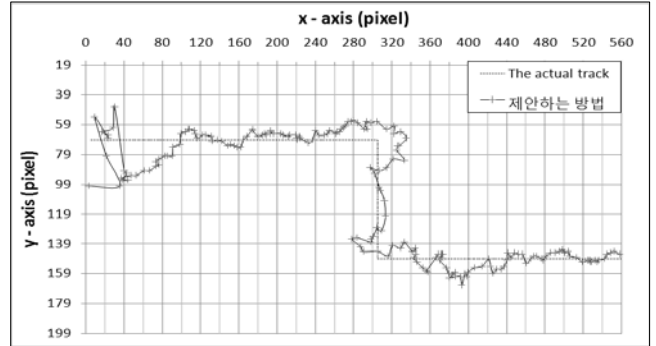
(그림 8) 제안하는 방법의 이벤트 흐름도

5. 정확도 제고 방법

Trilateration을 사용하는 것보다 EKF를 사용할 경우에 실제 경로에 더 가깝게 실험 결과가 나왔다. 그리고 파라메타 값에 따라 EKF 추적 결과도 차이가 있음을 알 수 있었다. 평균 오차에서는 실험 1이 실험 2보다 좋았지만, 추적 경로 중 직선경로에서 만큼은 실험 2가 실험 1보다 완만한 형태의 결과로 더 좋았다. 즉, 보행자가 일정한 방향으로 진행되는 동안에는 R_{value}=1, Q_{value}=0.00001이 더 바람직한 결과를 보이는 반면, 진행 방향이 바뀌는 영역에서는 R_{value}=1, Q_{value}=0.001이 더 바람직한 결과를 낳는다.

따라서 본 논문이 제안하는 EKF의 정확도를 제고하는 방안은 도면 정보를 이용하여 갈림길 영역을 파악하고, 갈림길 영역에서는 R_{value}=1, Q_{value}=0.001을 적용하고, 그 외의 영역에서는 R_{value}=1, Q_{value}=0.00001을 적용하는 것이다. 그림 8은 제안하는 방법의 이벤트 흐름도이다.

제안하는 방법을 4절 실험 데이터에 적용한 결과가 그림 9에 보인다. 실제 경로와 실험 결과로 얻은 경로의 평균 오차는 1.54m이다.



(그림 9) 제안하는 알고리즘을 적용한 실험 결과
<표 3> 각 실험에 따른 평균 오차 비교 (단위 m)

	Trilateration	실험1	실험2	제안하는 방법
평균오차	3.45	1.81	2.33	1.54

6. 결론

본 연구에서는 사용자가 복도를 이동하며 Δt 주기마다 신호세기를 측정하여 propagation model을 통해 AP까지의 거리들을 구하고, 이 거리들을 가지고 Trilateration을 적용하는 추적 시스템과 EKF를 적용하는 추적 시스템을 구현하였다. 또한 EKF를 이용한 추적 시스템에서 파라메타를 달리하며 실험함으로써 표 3과 같은 결과를 얻었다.

따라서 EKF의 정확도를 제고하기 위하여 도면 정보를 이용하여 갈림길 영역을 파악하고, 갈림길 영역에서는 R_{value}=1, Q_{value}=0.001을 적용하고, 그 외의 영역에서는 R_{value}=1, Q_{value}=0.00001을 적용하는 방안을 제안하였다. 이 경우 경로에 대한 평균오차가 1.54m로 더 바람직한 결과를 낳았다. 또한 상태 예측치 \hat{x}_{k+1} 의 공분산 행렬인 P_{k+1} 의 요소가 상수(작은 값)로 수렴함으로써 예측이 잘 되었음을 알 수 있었다.

본 연구에서는 사용자의 위치에서 AP까지의 거리들을 k번째까지 취득한 후에 현재 위치를 예측하였는데, 향후에는 k+1번째 이후의 거리들을 취득하지 않고서도 k+1번째 이후의 위치를 예측하는 방법을 연구하고자 한다. 또한 본 시스템에서 필요로 하는 도면 정보를 AutoCad 파일로부터 얻고자 한다.

참고문헌

- [1] A. Kotanen, M. Hannikainen, H. Leppakoski, T.D. Hamalainen, "Experiments on Local Positioning with Bluetooth", Proceedings of the International Conference on Information Technology: Computers and Communications, IEEE, April 2003, P. 297-303
- [2] F. Lassabe, P. Canalda, P. Chatonnay, F. Spies, "A Friis-based calibrated model for WiFi terminals positioning," Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. 13-16 June 2005, pp. 382-387
- [3] Maria-Gabriella Di Benedetto and Guerino Giancola, Understanding Ultra Wide Band: Radio Fundamentals, Prentice Hall, p. 431