

지문방식 측위 기반 칼만필터 추적의 정확성 제고 방법

임재걸, 정승환^o

동국대학교 컴퓨터멀티미디어학과
{yim, honourj^o}@dongguk.ac.kr

Techniques to Improve Accuracy of Fingerprinting-Positioning-Based Kalman Filter Tracking

Jaegeol Yim, Seunghwan Jeong^o

Dept of Computer and Multimedia, Dongguk University

요 약

위치기반서비스에서 사용자의 정확한 위치가 요구되면서 측위와 추적에 대한 연구가 활발히 진행되고 있다. 측위 방법에는 위성기반 방법[1, 2], 로컬네트워크기반 방법[3 - 6], 센서기반 방법[1, 7, 8, 9] 등이 있다. 본 연구에서는 로컬네트워크 중 WLAN (Wireless Local Area Network) 환경의 옥내에서 사용자의 위치를 추적하는 기존의 방법의 정확성을 제고하는 방안을 제안한다. 제안하는 방법은 WLAN 환경에서 RSS를 측정하여 K-NN방식으로 현재 위치를 판단한 다음, 칼만필터를 사용하여 사용자의 위치와 이동 경로를 예측한다는 점에서 기존의 방법과 비슷하다. 제안하는 방법의 특징은 도면 정보를 이용하는 것이다.

제안하는 방법은 도면정보로부터 갈림길 영역을 파악하고, 갈림길 영역에서는 측정치에 가중치를 두고 갈림길이 아닌 지역에서는 시스템 모델에 가중치를 두도록 파라메타의 값을 조절한다. 제안하는 방법의 효율성을 실험적으로 증명하기 위한 실험 결과와 분석 내용도 제시한다.

1. 서론

본 연구에서는 무선로컬네트워크(WLAN) 환경의 옥내에서 사용자의 이동 경로를 추적하는 칼만필터 방법을 살펴본다. 이동경로는 일정 시간이 경과한 때 시점에 측정된 사용자의 위치로 구할 수 있는데, 이렇게 구한 경로는 측위 방법의 오류 때문에 실제 경로와 차이가 있다. 본 연구에서는 칼만필터를 이용하여 이 차이를 극소화하는 방법을 제시한다.

WLAN 환경의 옥내 측위 방법들 중 가장 정확도가 높은 방법은 K-NN 방법으로 알려져 있다[3]. K-NN 방식은 준비단계(offline phase)와 실시간 단계(online phase)로 구성된다. 준비단계에서는 측위 소프트웨어가 적용되는 영역을 바둑판 모양으로 나누고, 두 선이 교차하는 지점들을 후보지점이라고 한다. 그리고 각 후보지점에서 AP(Access Point: WLAN에 설치된 접근장치)들의 신호의 세기(RSS: Received Signal Strength)를 측정할 것을 지문이라 한다. 준비단계에서는 후보지점과 지문의 쌍으로 구성된 찾아보기 표를 작성한다. 실시간 측정 단계에서는 사용자의 현재 위치에서 측정된 지문과 찾아보기 표의 지문들을 비교하여 가장 유사한 K 개의 후보지점들을 선택하고, 이들의 평균 좌표를 사용자의 위치로 판정한다.

K-NN 측위는 WLAN 환경에서 일반적으로 2-3m의 오차를 보임으로 더욱 정확한 경로를 작성하기 위하여

칼만필터를 사용한 결과가 [10]에 보인다. 본 연구에서도 [10]과 같이 K-NN 측위 방법을 사용하고 칼만필터로 경로를 작성한다. 단, [10]에서는 사용자의 위치를 2차원으로 계산하였는데 본 연구에서는 사용자의 위치를 3차원으로 확장하고, 필터링에 사용되는 파라메타들에 따라 달라지는 실험 결과들을 비교 분석하여 적절한 파라메타 값을 찾고, 나아가서 파라메타 값을 적절히 적용함으로써 경로의 정확성을 제고하는 방법을 제시한다.

2. 기존의 연구

본 절에서는 기존의 측위 방법 중 K-NN 방법과 추적 방법 중 칼만필터 방법을 소개한다.

2.1 K-NN 방법

K-NN 방법은 준비단계(offline phase)와 실시간 측정 단계(online phase)로 구성된다. 준비단계에서는 측위 소프트웨어가 적용되는 범위내의 각 후보지점에서 RSS를 측정하여 훈련데이터(training data)라 한다. 각 후보지점에서 일반적으로 3 개 이상의 AP들의 RSS가 읽힘으로 한 훈련데이터는 3개 이상의 RSS로 구성된다. 각 후보지점에서 100개 이상의 훈련데이터를 수집하여, 각 AP마다 훈련데이터의 RSS들의 평균을 구한다. K-NN 방법에서는

이와 같이 구한 훈련데이터의 평균을 지문이라 한다. K-NN 방법은 준비단계에서 후보지점 이름과 지문으로 구성된 찾아보기 표를 작성한다. 실시간 측정단계에서는 사용자의 현재 위치에서 측정한 RSS 값과 찾아보기 표의 내용을 비교하여 사용자의 위치를 판정한다. 이때 K 개의 가장 비슷한 항목을 찾아, 이 후보지점들의 좌표의 평균을 사용자의 현재 위치로 판정하는 방법을 K-NN이라 한다.

예를 들어 후보지점 A, B, C, ... 각각에서 여러 번 측정한 RSS의 평균값들로 구성된 찾아보기 표의 일부가 표 1과 같다고 하자. 표에서 RSS_i는 AP_i가 발신한 신호의 세기이다.

표 1. 찾아보기 표의 예

후보지점	RSS ₁	RSS ₂	RSS ₃
A	31	48	55
B	37	47	54
C	40	45	52
D	45	45	50
...

실시간 단계에서 사용자의 현재 위치에서 RSS 값을 실측하고, 실측한 값과 가장 가까운 항목을 찾아보기 표에서 K개 찾는다. 예를 들어 실시간 측위 단계에서 실측한 값이 X=(38, 46, 53)이라 하면, X와 A의 거리는 $\sqrt{(38-31)^2 + (46-48)^2 + (53-55)^2}$ 이 된다. 비슷한 방법으로 다른 후보지점들에 대한 거리도 구한다. K가 1이라면 X와 가장 가까운 후보지점 하나를 찾아 현재 위치가 그곳이라고 판정한다. K=2라면 가장 가까운 두 후보지점을 찾아 이들의 평균(지점이 좌표임으로 좌표의 평균)을 현재 지점으로 판정한다.

2.2 칼만필터

[10]에 소개된 칼만필터를 이용한 추적 방법은 다음과 같다. 칼만필터는 상태 $X(k)$ 와 측정 $Z(k)$ 에 대한 선형적 확률차분방정식을 사용하여 이산 시간으로 제어되는 처리과정을 모델 한다.

$$X(k) = AX(k-1) + Bu(k) + n(k-1)$$

$$Z(k) = HX(k) + v(k)$$

여기서 A, B, H 행렬은 선형 추이와 측정 처리 과정을

정의하며, 무작위 벡터 n 은 진행 잡음, 무작위 벡터 v 는 측정치 잡음을 나타낸다. 이들은 다음과 같이 서로 독립인 백색 정규 분포로 가정된다.

$$p(n) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

여기서 Q 는 상태 오차의 공분산 행렬이고 R 은 측정치에러의 공분산 행렬이다. K-NN 측위 기반의 경로 추적 시스템을 위해, 다음과 같은 파라메타들을 정의한다.

$$X(k) = \begin{bmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$Z(k) = \begin{bmatrix} x_{RSS}(k) \\ y_{RSS}(k) \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

여기서 x, y 는 x축, y축의 위치이고, v_x, v_y 는 x축, y축 방향의 속도이며, Z 는 Δt 주기로 취득한 K-NN 측위로 계산된 위치이다. 그러면 예측 상태 \hat{X} 은 다음과 같이 위치 '예상'과 위치 '보정'의 단계[1, 11]를 사용하여 계산 될 수 있다.

$$\hat{X}^-(k) = A\hat{X}(k-1)$$

$$P^-(k) = AP(k-1)A^T + Q$$

$$K(k) = P^-(k)H^T(HP^-(k)H^T + R)^{-1}$$

$$\hat{X}(k) = \hat{X}^-(k) + K(k)(Z(k) - H\hat{X}^-(k))$$

$$P(k) = (I - K(k)H)P^-(k)$$

여기서 $P^-(k)$ 는 k번째 선형적 상태 예측치의 공분산 행렬이고 $P(k)$ 는 k번째 측정치가 반영된 k번째 후협적 상태 예측치의 공분산 행렬이다.

3. 기존의 사용자 추적 시스템 구현

[10]에 소개된 칼만필터를 이용한 사용자 경로 추적 시스템은 K-NN 방법을 적용한 측위 모듈과 칼만필터를 적용한 추적 모듈로 구성된다. 구현한 측위 모듈은 기존의 연구에서 소개한 K-NN 방법임으로 지면 제한으로 구현에 대한 설명을 생략하고 칼만필터 부분에 대한 구현 사항을 소개한다. Intel(R) PRO/Wireless 2200BG Network Connection이 랜 카드로 장착된 Lab-Top 컴퓨터에서

Microsoft Visual Studio .NET 2005의 C#언어를 사용하여 시스템을 구현한다.

3.1 칼만 필터 사용자 추적 모듈

2절에 소개된 사용자 추적을 위한 칼만필터를 구현한다. 단, $X(k)$ 를 3차원 좌표로 확장하여, 사용되는 파라메타 들을 다음과 같이 정의한다.

$$X(k) = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \\ v_x(k) \\ v_y(k) \\ v_z(k) \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Z(k) = \begin{bmatrix} x_{RSS}(k) \\ y_{RSS}(k) \\ z_{RSS}(k) \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

여기서 A, H 행렬은 선형 추이와 측정 처리 과정을 정의하며, x, y, z 는 x축, y축, z축의 위치이고, v_x, v_y, v_z 는 x축, y축, z축 방향의 속도이다. $Z(k)$ 는 Δt 주기로 K-NN 측위 결과이다.

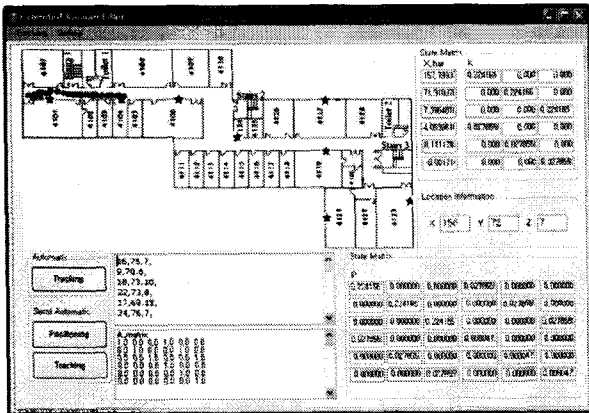


그림 1. 사용자 추적 시스템의 실행화면

3.2 구현 결과

그림 1에 구현한 추적 시스템이 보이는데, 이 응용프로그램은 Tracking과 Setting의 두 개의 메뉴로 구성되어 있다. Setting 메뉴에서 칼만필터의 초기 파라메타 값들을 설정한 다음 Tracking 메뉴에서 사용자의 위치를 추적하게 된다. Tracking 메뉴에서는 지도와 Location Information을 이용하여 위치와 경로가 표시되며, Δt 주기마다 행렬

$\hat{X}(k), K(k), P(k)$ 의 값이 갱신된다. Semi Automatic 그룹박스 안에 있는 두 개의 버튼 중 Positioning 버튼을 누르면 현재 위치를 K-NN 측위 모듈로 계산하며, 그 다음 Tracking 버튼을 누르면 칼만필터를 이용하여 사용자의 위치를 예측하게 된다. 이 그룹박스는 사용자가 임의적으로 위치를 추적하는 방법으로써 Δt 주기마다 자동으로 위치를 추적할 경우엔 Automatic 그룹박스의 Tracking 버튼을 사용한다.

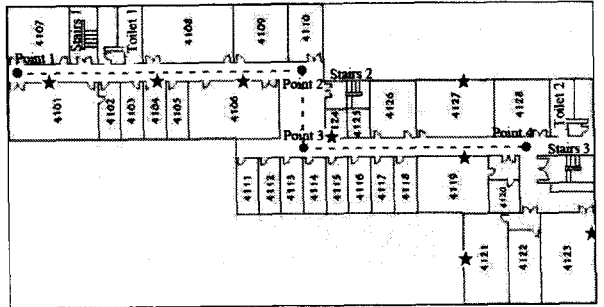


그림 2. 자연과학관 4층

4. 실험

본 연구에서는 WLAN 환경의 IEEE 802.11b 규격의 2.4GHz RF의 RSS를 이용하여 그림 2에 보이는 자연과학관 4층(85.5m x 43.8m) 복도에서 사용자 추적 시스템을 실험하였다.

4.1 실험 환경

실험에 사용되는 자연과학관 4층의 AP들은 그림 2에 '★'로 표시된 AP1부터 AP8까지 총 8개가 사용되었으며, 이들의 지도상의 3차원 좌표는 표 2와 같다. 여기에 사용된 1픽셀의 실제 거리는 약 0.1407m이다.

표 2. 자연과학관 4층 AP들의 지도상의 3차원 좌표

AP index	3차원 좌표 (x축, y축, z축)
AP1	52, 75, 19
AP2	156, 75, 19
AP3	245, 79, 19
AP4	330, 136, 19
AP5	472, 80, 19
AP6	472, 160, 19
AP7	472, 263, 19
AP8	605, 227, 19

사용자의 이동 경로는 그림 2의 점선이며 좌측 상단의 Point 1을 시작으로 우측 중앙의 Point 4까지로 한다. 사용자는 복도를 이동하며 Δt 주기로 K-NN 측위 결과를 구하여 사용자 추적 시스템의 $Z(k)$ 로 사용해서 예측 상태 \hat{X} 를 구한다. 여기서 $3\Delta t$ 는 1초이다.

실제 경로를 걸어가면서 매초 3회씩 K-NN 측위하는 실험을 100회 반복하여 K-NN 측위의 평균을 구한 결과를 실제 경로와 함께 나타낸 그림은 그림 3과 같으며, 이 경우 평균 오차는 1.51m이다.

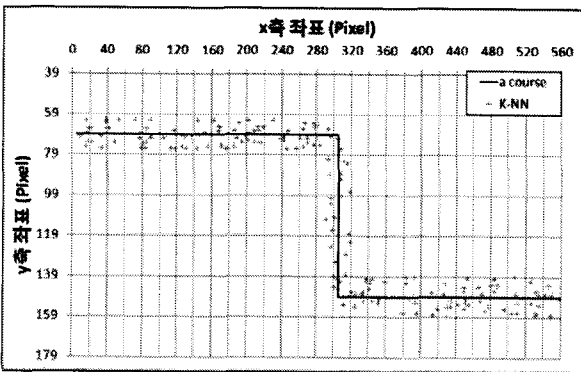


그림 3. 실제 경로와 K-NN 결과

4.2 실험 결과

사용자 추적 시스템의 초기 파라메타는 다음과 같이 동일하게 설정하였다.

$$\hat{X}_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$P_0 = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$$

그리고 상태 오차 Q (6×6 matrix)와 측정치 오차 R (3×3 matrix)는 대각행렬이면서 대각선상의 원소들의 값이 동일한 스칼라행렬이다. 여기서 스칼라행렬 R 과 Q 의 대각선상의 원소의 값을 R_{value} 와 Q_{value} 라 한다. 그 다음 칼만필터의 파라메타를 표 3과 같이 설정하여 실험한 결과와 각 실험에서의 $\hat{X}(k)$ 의 공분산 행렬인 $P(k)$ 의 변화를 살펴보았다.

표 3. 스칼라대각행렬 R 과 Q 의 대각선상의 원소 값 설정

실험 index	R 의 대각선상의 원소 R_{value}	Q 의 대각선상의 원소 Q_{value}
실험 1	0.001	1
실험 2	1	0.001

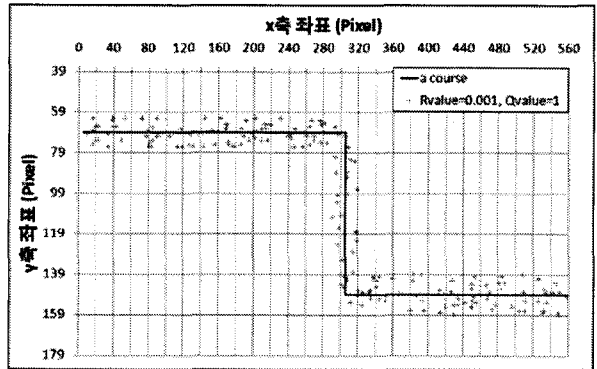


그림 4. EKF의 파라메타를 실험 1로 설정한 경우

R_{value} 가 Q_{value} 보다 훨씬 작은 표 3의 실험 1의 결과가 그림 4와 같이 나타났다. 이 경우에는 칼만필터의 결과가 K-NN의 결과와 동일함을 볼 수 있으며, 평균 오차도 동일하게 1.51m가 나왔다.

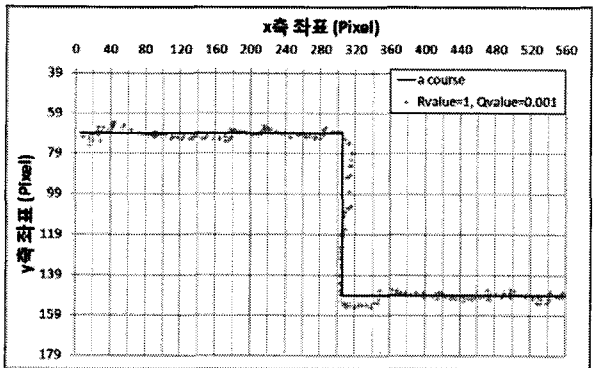


그림 5. EKF의 파라메타를 실험 2로 설정한 경우

그림 5는 R_{value} 가 Q_{value} 보다 훨씬 큰 표 3의 실험 2의 경우로, 직진하는 경로의 추적 결과가 사용자의 이동 경로에 근접하게 나왔다. 이 경우의 평균 오차는 0.77m가 나왔다.

$\hat{X}(k)$ 의 공분산 행렬 $P(k)$ 의 요소를 다음과 같이 $value - a$, $value - b$, $value - c$ 로 정의하여

$$P_k = \begin{bmatrix} value-a & 0 & 0 & value-b & 0 & 0 \\ 0 & value-a & 0 & 0 & value-b & 0 \\ 0 & 0 & value-a & 0 & 0 & value-b \\ value-b & 0 & 0 & value-c & 0 & 0 \\ 0 & value-b & 0 & 0 & value-c & 0 \\ 0 & 0 & value-b & 0 & 0 & value-c \end{bmatrix}$$

실험 2가 진행될 때 $P(k)$ 의 값들을 살펴보면 그림 6과 같다. 추적이 진행될수록 $P(k)$ 의 요소가 $value-a = 0.2241$, $value-b = 0.0279$, $value-c = 0.008$ 의 작은 값(상수)으로 수렴되는 것을 볼 수 있다.

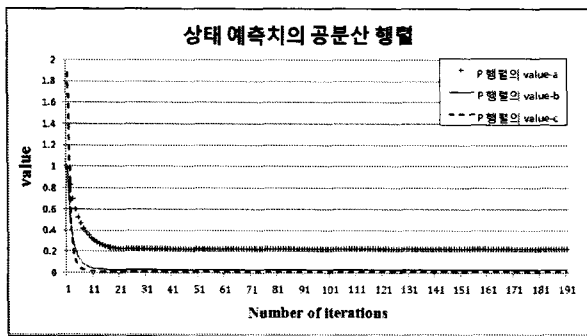


그림 6. 상태 예측치 $\hat{X}(k)$ 의 공분산 행렬인 $P(k)$ 의 변화

5. 정확도 제고 방법

실험 결과 칼만필터를 사용하면 K-NN 측정치를 그대로 사용하는 것보다 더 실제 경로에 가깝다는 것을 알 수 있었으며, 파라메타 값에 따라 칼만필터 추적 결과도 차이가 많음을 알 수 있었다. 즉, 측정오차와 시스템오차를 각각 $R_{value}=0.001$, $Q_{value}=1$ 로 배정할 경우에는 추적 결과가 K-NN 측위 결과와 같았고, $R_{value}=1$, $Q_{value}=0.001$ 로 배정할 경우에는 추적 결과가 직선에 가까워짐을 보였다. 따라서 보행자가 일정한 방향으로 진행하는 동안에는 $R_{value}=1$, $Q_{value}=0.001$ 이 더 바람직한 결과를 보이는 반면, 진행 방향이 바뀌는 영역에서는 $R_{value}=0.001$, $Q_{value}=1$ 이 더 바람직한 결과를 낸다.

보행자의 진행 방향이 바뀔 수 있는 지점은 갈림길이나 두 개의 길이 교차하는 지점으로 이러한 지점들은 도면 정보로 파악할 수 있다. 예를 들어 그림 2의 경우에는 갈림길이 나타나는 지점의 영역이 다음과 같이 두 영역 (도면상의 Pixel)이다.

- 영역1: $280 \leq x \leq 328$ and $60 \leq y \leq 97$
- 영역2: $280 \leq x \leq 328$ and $138 \leq y \leq 159$

따라서 본 논문이 제안하는 칼만필터의 정확도를 제고 하는 방안은 도면 정보를 이용하여 갈림길 영역을 파악 하고, 갈림길 영역에서는 $R_{value}=0.001$, $Q_{value}=1$ 을 적용 하고, 그 외의 영역에서는 $R_{value}=1$, $Q_{value}=0.001$ 을 적용 하는 것이다. 그림 7은 제안하는 방법의 이벤트 흐름도 이다.

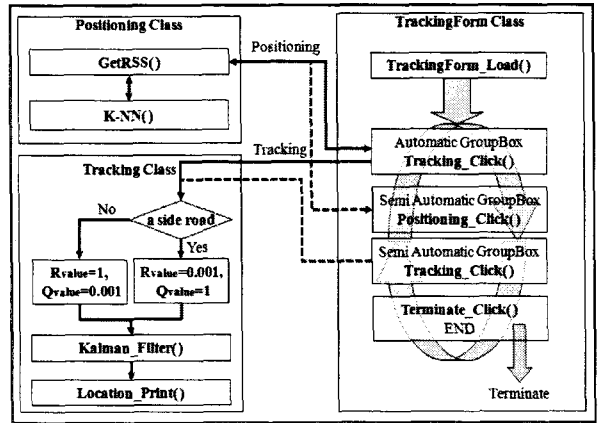


그림 7. 제안하는 방법의 이벤트 흐름도

제안하는 방법을 4절 실험 데이터에 적용한 결과가 그림 8에 보인다. 실제 경로와 실험 결과로 얻은 경로의 평균오차는 0.67m이다.

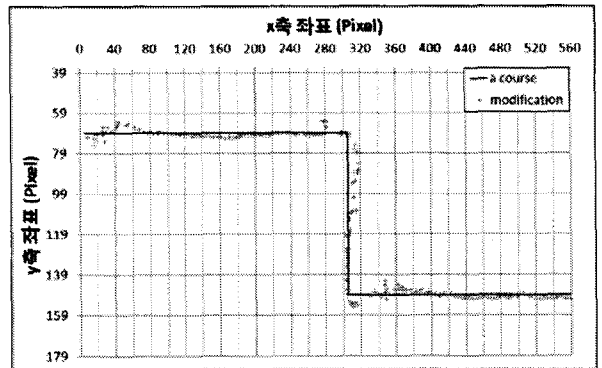


그림 8. 제안하는 알고리즘을 적용한 실험 결과

K-NN 측정치로 작성한 경로의 평균 오차, $R_{value}=0.001$, $Q_{value}=1$ 인 칼만필터로 작성한 경로의 평균오차, $R_{value}=1$, $Q_{value}=0.001$ 인 칼만필터로 작성한 경로의 평균 오차, 제안하는 알고리즘으로 작성한 경로의 평균오차를 하나의 표로 보이면 표 4와 같다.

표 4. 각 실험에 따른 평균오차 비교 (단위 m)

	K-NN	$R_{value}=0.001,$ $Q_{value}=1$	$R_{value}=1,$ $Q_{value}=0.001$	제안하는 알고리즘
평균 오차	1.51	1.51	0.77	0.67

6. 결론

본 연구에서는 K-NN 측위 방법과 [10]에 소개된 칼만 필터 추적 방법을 적용하여 사용자 추적 시스템을 구현하였다. 이때 [10]과는 달리 사용자의 위치 정보를 2차원에서 3차원으로 확장하였다.

칼만필터의 파라메타를 달리하며 실험한 결과 $R_{value}=0.001, Q_{value}=1$ 로 배정할 경우에는 추적 결과가 K-NN 측위 결과와 같았고 평균오차는 1.51m이었다. 그리고 $R_{value}=1, Q_{value}=0.001$ 로 배정할 경우에는 평균오차가 0.77m로 추적 결과가 직선에 가까웠다. 따라서 칼만필터의 정확도를 제고하기 위하여 도면 정보를 이용하여 갈림길 영역을 파악하고, 갈림길 영역에서는 $R_{value}=0.001, Q_{value}=1$ 을 적용하고, 그 외의 영역에서는 $R_{value}=1, Q_{value}=0.001$ 을 적용하는 방안을 제안하였다. 이 경우 평균오차가 0.67m로 더 바람직한 결과를 낳았다. 또한 그림 6처럼 상태 예측치 $\hat{X}(k)$ 의 공분산 행렬인 $P(k)$ 의 요소가 상수(작은 값)로 수렴함으로써 예측이 잘 되었음을 알 수 있었다.

본 연구에서는 k번째까지 취득한 K-NN 결과를 가지고 현재 위치를 예측하였는데, 향후에는 k+1번째 이후의 K-NN 결과를 취득하지 않고서도 k+1번째 이후의 위치를 예측하는 방법을 연구하고자 한다.

참고문헌

[1] D. Obradovic, H. Lenz and M. Schupfner, "Sensor Fusion in Siemens Car Navigation System," IEEE Conference on Machine Learning and Signal Processing, São Luis, Brazil, 2004. 6

[2] Dixon, T.H., "An Introduction to the Global Positioning System and Some Geological Applications," Reveiw of Geophysics, vol. 29, p. 249-276, May 1991.

[3] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based User Location and Tracking

system," in proceeding of IEEE INFOCOM 2000, 2(3), pp. 775-784, March 2000.

[4] Ingram, S.J. Harmer, D. Quinlan, M., "Ultra-Wide band Indoor Positioning Systems and Their Use in Emergencies," in proceedings of Position Location and Navigation Symposium 2004 (PLANS 2004), pp. 706-715, April 2004.

[5] B.B. Parodi, H. Lenz, A. Szabo, H. Wang, J. Horn, J. Bamberger and J. Obradovic, "Initialization and Online-Learning of RSS Maps for Indoor/Campus Localization," IEEE/ION PLANS 2006, San Diego, USA, April, 2006.

[6] M. Vossiek, L. Wiebking, P. Gulden, J. Wiegardt, C. Hoffmann, "Wireless local positioning - concepts, solutions, application," IEEE Microwave Magazine, vol. 4, pp. 77-87, December 2003.

[7] R. Want, A. Hopper, V. Falcão and J. Gibbons, "The active badge location system," ACM Transactions on Information System, 10(1), pp. 91-102, January 1992.

[8] N. B. Priyantha, A. Chakraborty, H. Balakrishnan, "The Cricket Location-Support System," in proceedings of MOBICOM 2000, pp. 32-43, August 2000.

[9] Hui Wang, Henning Lenz, Andrei Szabo, Joachim Bamberger, Uwe D. Hanebeck, "Fusion of Barometric Sensors, WLAN Signals and Building Information for 3-D Indoor/Campus Localization," in proceeding of International Conference on Multi-Sensor Fusion and Integration for Intelligent Systems (MFI), 2006, Heidelberg, Germany.

[10] H. Wang, H. Lenz, A. Szabo, J. Bamberger, U. D. Hanebeck, "WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors", Workshop on Positioning, Navigation and Communication, (2007 WPNC), Hannover, Germany, March 2007.

[11] Branko Ristic, Sanjeev Arulampalam, Neil Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House Publishers, February 2004.