

위치기반서비스를 위한 지도정보가 반영된 옥내측위통합 시스템

임재걸* · 주재훈** · 정승환***

〈 목 차 〉

I. 서론	4.3 제어부분
II. 관련연구	V. 통합 시스템 구현 및 실험
III. 위치추적 알고리즘	5.1 통합시스템 구현
3.1 EKF	5.2 실험
3.2 WLAN 기반 EKF 추적 알고리즘	VI. 결론
IV. 지도보기 프로그램	참고문헌
4.1 입력부분	부록 A: DXF 파일 형식
4.2 그리기 부분	

I. 서론

국내에서도 이미 이동통신회사를 중심으로 다양한 위치기반 모바일 서비스(SKTEL의 Nate Drive, KTF의 친구찾기, LGT 애인안심서비스 등)가 상용화되었다. 위치기반서비스(Location Based Service: LBS)는 안전 및 구조 서비스, 주변정보제공 서비스, 추적 서비스, 교통 및 항법 서비스, 위치기반상거래 서비스 등 다양한 분야에 적용되고 있다(김태성 등, 2006). 또한 오늘

날 LBS는 단말기 제조산업, 콘텐츠산업, 통신산업 등의 성장에 커다란 영향을 미치기 때문에 중요한 연구 분야이다(박종현 등, 2003).

LBS를 위한 핵심기술 분야는 위치측위기술이다. 측위기술은 옥외와 옥내 측위방식으로 구분된다. 오늘날 상용 위치기반서비스가 기업은 물론 일상생활에도 널리 사용되고 있는데, 여기서서는 모두 옥외 측위방식을 활용하고 있다. 옥외에서는 GPS(Global Positioning System)를 이용한 측위 방식(Han and Xue, 2004; Gehrig et al., 2003; Yang et al., 2003)이 보편화되어 있기 때문

* 동국대학교 컴퓨터멀티미디어학과 교수, yim@dongguk.ac.kr

** 동국대학교 경영·관광대학 전자상거래학과 교수, givej@dongguk.ac.kr

*** 동국대학교 연구원

에 이것을 응용한 상용 위치기반서비스용 시스템(Gehrig et al., 2003; Yang et al., 2003)이 활발하게 개발되어 왔다. 그러나 옥내의 경우에는 보편적인 측위 기술 개발이 미미한 상황이다(임재걸 등, 2006). 건물 내에서는나 장애물이 많은 곳에서는 GPS 수신기 불가능함으로 옥내 측위가 필수적이다. 또한 사용자의 편리성과 현실적 관점에서 볼 때, 하나의 단말기로 옥외에서는 물론이고 옥내에서도 위치기반서비스가 가능하도록 할 필요가 있기 때문에 옥내 측위방식에 대한 연구도 중요하다.

본 연구의 목적은 옥내측위 방식을 제안하고 지도보기 모듈을 개발하고, 나아가서 옥내측위 방식에 지도보기 프로그램을 통합하는 시스템을 개발하여 상용 옥내 위치기반서비스 시스템 개발의 디딤돌을 마련하는 것이다.

본 연구에서 개발하는 옥내측위 소프트웨어는 무선근거리통신망(Wireless Local Area Network: WLAN) 환경에서 사용자의 위치를 측정한다. 다양한 옥내측위 방법 중, WLAN 기반 옥내측위는 측위만을 위한 특수 장비를 요구하지 않기 때문에 경제적이라는 장점이 있다. WLAN 기반 측위의 목적으로 사용될 수 있는 기술이 다양하게 발표된 바 있는데, 본 연구에서는 여러 AP(Access Point)의 신호세기(Received Signal Strength: RSS)를 측정한 후 거리와 신호세기간의 관계식(propagation model이라 함)(Lassabe, 2005)을 이용하여 사용자의 위치로부터 AP까지의 거리를 계산하고, 구한 거리와 AP의 위치를 바탕으로 사용자의 위치를 측정한다. 이때 사용자가 움직이는 습성을 바탕으로 사용자의 위치를 추정하고, 계산한 거리로 추정한 위치를 교정하고, 또 다시 추정하는 과

정을 반복함으로써 사용자의 위치를 판단하는 EKF(Extended Kalman Filter) 방법을 적용한다(Gelb, 1974; Hwang and Brown, 1992; Kalman, 1960; Maybeck, 1982;).

본 연구에서 개발하는 지도보기 소프트웨어는 오토캐드(Finkelstein, 2007)로 작성한 DXF(Drawing Exchange Format) 형식의 옥내 지도 파일을 입력으로 받아 옥내측위 시스템의 사용자 인터페이스에 출력한다. DXF 파일 형식은 호환성을 중요시한 Autocad 파일 형식으로 어느 프로그램에서도 읽을 수 있다는 특징이 있다. 또한 본 지도보기 소프트웨어는 출력한 지도를 확대·축소·이동하는 기능도 제공하며, 나아가서 옥내측위 소프트웨어가 제공하는 사용자의 위치를 지도상에 표시하는 기능도 제공한다.

본 논문은 다음과 같이 구성된다. II에서는 관련연구를 요약하고, III에서는 본 연구가 제안하는 위치추적을 위한 EKF 알고리즘을 제안하며, IV에서는 지도보기 모듈을 설계한다. V절에서는 통합 시스템과 실험결과를 제시하고, VI절에서 본 연구 결과를 요약하고 향후 연구 계획을 소개한다.

II. 관련연구

본 절에서는 측위에 대한 기존의 연구 결과를 소개하고, 지도보기 소프트웨어에 대하여 간단히 언급한다. 대표적인 측위 시스템으로 GPS와 GLONASS가 널리 사용된다(주재순 등, 2001). GPS는 미국이, GLONASS는 러시아가 제공하는 인공위성 기반 측위데이터 제공 시스템이다. 측위 위성은 항상 지구를 향하여 전파를

발사하고 있으며, GPS 수신 장치는 이 전파를 참조하여 위성에서 송신한 신호가 수신 장치에 도달하는데 걸리는 시간을 계산하고, 이 시간에 광속을 곱하여 수신 장치까지의 거리를 계산한다. 그러나 GPS 수신기는 위성과 수신장치 간에 아무런 장애물이 없어야만(Line-of-Sight) 측위 위성 데이터의 수신이 가능하다. 따라서 건물 내부에 있는 사용자의 위치를 파악하는데 GPS 수신기는 거의 아무런 도움을 주지 못한다. 그래서 측위는 옥외 측위와 옥내 측위로 구분되고, 옥외 측위는 GPS 방법을 중심으로 발전하고 있으며, 옥내 측위는 옥외 측위와 별도로 발전하고 있다. 옥내 측위는 옥내 위치기반서비스 시스템 구축에 열쇠가 되는 기술이다

다양한 옥내 측위 시스템이 발표되었다. 예를 들면, Active Badge(Want, 1992) 시스템에서는 이동물체들에 적외선을 발사하는 송신기가 부착되어 있고, 방마다 여러 개의 수신기가 고정 지점에 설치되어 있다. 적외선은 벽을 통과하지 못하기 때문에 이동물체가 어느 방에 들어오면 그 방에 설치된 고정 수신기들만 그 이동물체의 적외선을 감지할 수 있다. 수신기는 중앙컴퓨터에 연결되어 있어서 수신기에 감지된 적외선을 바탕으로 이동물체가 어느 방 안에 있다는 것을 판정한다. 이상에서 본 바와 같이, Active Badge 시스템에서는 측위를 위한 특수 장비가 필요하고, 측정된 위치의 단위가 방이라는 것을 알 수 있다.

BAT(Harter, 1997) 시스템에서는 이동물체에 부착된 송신기가 고주파(RF: Radio frequency) 신호와 초음파(Ultrasound)를 송신한다. 수신기들은 천정에 행렬 모양으로 고정되어 있다. 송신기가 고주파 신호와 초음파 신호를 동시에 송

신하면, 수신기는 이 두 가지 신호가 도착한 시각의 차이를 바탕으로 송신기와 수신기 사이의 거리를 구한다. 번개가 보인 시각과 천둥이 들린 시각의 차이로 번개가 친 장소와 나와의 거리를 구할 수 있는 것과 동일한 원리를 적용하는 것이다. 수신기들은 모두 중앙컴퓨터에 연결되어 있어서 중앙컴퓨터는 수신기들의 좌표와 이동물체와의 거리를 바탕으로 이동물체의 정확한 위치를 판정한다. 이상에서 본 바와 같이, BAT 시스템도 Active Badge와 마찬가지로 측위를 위한 특수 장비를 사용함으로써 설치 경비가 많이 든다는 것을 알 수 있다.

Cricket(Priyanthat et. al. 2000) 시스템에서는 Active Badge나 BAT와 반대로 이동물체 자신이 자신의 위치를 계산한다. Cricket도 BAT와 마찬가지로 고주파 신호와 초음파 신호를 함께 사용하는데, BAT에서는 이동물체에 부착된 송신기가 신호들을 송신한 반면, Cricket에서는 고정 지점에 부착된 비컨 (beacon)들이 고주파 신호와 초음파 신호들을 동시에 송신하면 이동물체에 부착된 수신기가 이 신호들을 수신하고, 신호들의 도착 시각의 차이를 바탕으로 거리들을 구하고, 비컨들의 좌표와 거리를 바탕으로 자신의 좌표를 구한다. 이상에서 본 바와 같이, Cricket 시스템도 역시 비컨과 수신기와 같은 측위를 위한 특수 장비를 사용한다.

RADAR(Bahl, 2000) 시스템에서는 이동물체가 고정 지점에 위치한 세 개의 기지 스테이션(base station: 탁상 컴퓨터를 사용함)에 UDP 패킷을 송신하면 기지 스테이션이 패킷의 신호 세기를 바탕으로 이동물체의 위치를 판독한다. 판독을 위하여 사용되는 기술은 K-NN (K nearest neighbor)이라는 일종의 지문(fingerprint) 방식이

다. 지문 방식은 준비단계 (off-line phase)와 실시간 측정 단계 (on-line phase)로 구성된다. 준비 단계에서는 측위 소프트웨어가 적용되는 범위 내의 각 지점(후보지점이라고 함)에서 특징 값 (RADAR에서는 UDP packet의 신호의 세기)을 측정하여, 측정치들의 평균으로 찾아보기 표를 작성한다. 실시간 측정 단계에서는 사용자의 현재 위치에서 측정된 특징 값과 찾아보기 표의 내용과 비교하여 사용자의 위치를 판정한다. 이때 K 개의 가장 가까운 지점을 찾아, 이들의 평균을 사용자의 현재 위치로 판정하는 방법을 K-NN이라 한다.

이상에서 살펴본 옥내 측위 방법들은 측위를 위한 특수 장비와 환경을 갖추어야 한다. 특수 장비를 갖추려면 상응하는 비용을 투자해야 할 뿐만 아니라 장비가 설치되어 있지 않은 곳에서는 적용할 수 없다는 단점이 있다. 따라서 측위를 위한 특수 장비가 전혀 필요 없는, 무선 랜을 이용하는 방식이 이미 여러 곳에서 발표된 바 있다.

Lassabe(Lassabe, 2005)는 AP(Access Point)와 이동물체와의 거리를 D, 이동물체에서 측정된 AP의 신호세기를 S라 하고, D와 S 사이의 상관 관계를 구하여 사용자의 위치를 판정하는 방법의 실험 결과를 발표한 바 있다. 그러나 이 방법은 Priyanthat(Priyanthat et. al. 2000)이 지적한 바와 같이 오차가 매우 크다.

측정치에 오차가 많을 경우에, 시스템 상태의 모형과 측정치를 조화롭게 조합하는 방법으로 칼만필터가 일반적으로 사용된다. EKF를 이용한 옥내 측위 방법이 이미 Kotanen, et al.,(2003)에 의하여 발표된 바 있다. 그러나 본 연구에서 개발하는 측위 방식은 Kotanen, et al.,(2003)의

방법과 다음과 같은 두 가지 점에서 크게 다르다. 첫째, 그들은 블루투스(bluetooth) 통신 환경의 측위인데 반하여 본 연구의 측위 환경은 WLAN이라는 점이다. 둘째, 그들 연구에서는 측위 방법만 다루는데 반하여 본 연구는 추적 방식도 다루고 있다는 점이다.

본 연구에서는 옥내측위 모듈과 더불어 옥내 지도보기 모듈도 개발한다. 지도보기 소프트웨어는 벡터 방식으로 표현된 지도정보를 입력으로 받아 사용자 인터페이스에 출력하고, 확대, 축소, 이동 등의 지도조작 기능을 제공한다. 이러한 지도보기 소프트웨어는 측위 시스템과 통합되어 사용자의 위치를 고려한 유용한 정보를 편리하게 보여준다. 이러한 기능을 하는 많은 옥외용 소프트웨어는 많이 개발되었지만 (Bruntrup et al., 2005), 옥내용으로 개발된 것은 찾아볼 수 없다.

Ⅲ. 위치추적 알고리즘

여기서는 먼저 기존의 확장칼만필터를 소개하고, 다음으로 WLAN 기반 추적 EKF 알고리즘을 제안한다.

3.1 EKF

1960년 R.E. Kalman(1960)의 논문을 시작으로 많은 연구가 진행된 칼만필터는 시스템의 상태와 측정치를 효율적으로 결합하는 방법으로, 상태를 예상하는 단계와 측정치를 적용하여 예상치를 조절하는 단계를 반복적으로 실행하여 시스템 상태 값을 계산한다. 칼만필터가 계산한

결과를 시스템 상태의 예측치라 하며, 칼만필터를 측위에 적용할 경우에는 사용자의 위치를 예측한다. 또한 본 연구에서는 측정치로 사용자의 현재 위치(사용자가 모바일 단말기를 소지하고 있음)에서 고정스테이션까지의 거리를 이용한다.

본 연구에서는 고정스테이션으로 무선근거리통신망용 AP를 이용한다. 따라서 측정 거리는 모바일 단말기에서 수신한 AP의 신호를 거리로 환산하여 구한다. AP의 신호에는 잡음이 많음으로 측정 거리의 오차가 큰 것도 필연적이다. 예측치가 좌표인데 반하여 측정치는 거리임으로 예상단계의 결과인 좌표와 측정 단계의 결과인 거리의 호환성을 위하여 서로 변환하는 과정이 본 연구의 측위를 위한 칼만필터에 반영되어야 하는데 이러한 칼만필터를 확장칼만필터(EKF)라고 한다.

모르는 위치의 모바일 단말기, $A = [x \ y \ z]^T$ 에서 알고 있는 위치의 AP, (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_i^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 AP의 위치와 기준점으로 계산한

$$\text{거리이며, } (h_x^i = \frac{X^i - x_0}{r_0^i}, h_y^i = \frac{Y^i - y_0}{r_0^i},$$

$$h_z^i = \frac{Z^i - z_0}{r_0^i}) \text{는 기준점에서 AP로의 시선}$$

(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 $X_{k+1} = X_k + W_k$, $Z_k = h(X_k) + V_k$ 이고, Kalman Filter를 적용하기 위하여 비선형 측정식

Linearized state model:

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

Linearized measurement model:

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

1) Initial guess:

$$\bar{x}_0^- = E(\bar{x}_0) \text{ and } P_0^- = \text{var}(\bar{x}_0)$$

2) Linearizing:

$$\bar{r}_k = \bar{r}_0 + H_k \delta \hat{x}_k + \bar{v}_k,$$

$$\bar{r}_0 = \begin{bmatrix} \sqrt{(X_1 - x_k^-)^2 + (Y_1 - y_k^-)^2 + (Z_1 - z_k^-)^2} \\ \vdots \\ \sqrt{(X_N - x_k^-)^2 + (Y_N - y_k^-)^2 + (Z_N - z_k^-)^2} \end{bmatrix}$$

3) Kalman Gain:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

4) Measurement update:

$$\delta \hat{x}_k = \delta \hat{x}_k^- + K_k (\bar{r}_k - \bar{r}_0^-), \hat{x}_k = \hat{x}_k^- + \delta \hat{x}_k$$

5) Update error covariance:

$$P_k = (I - K_k H_k) P_k^-$$

6) State propagation:

$$\hat{x}_{k+1}^- = \Phi_k \hat{x}_k, P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k$$

7) Goto step2

$\delta Z_k = H_k \delta X_k + V_k$ 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 EKF를 구성하면 최종 위치는 다음과 같은 식으로 구한다.

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

$$\hat{Z}_{k+1} = h(\hat{X}_k)$$

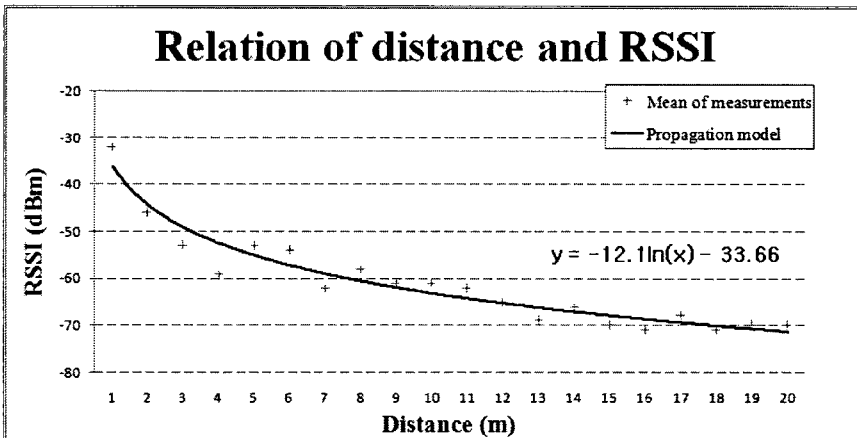
<그림 1>에서는 EKF의 시스템 모델과 측정 모델을 정리하여 최종 위치를 구하는 과정을 나타내었다. 여기서 \bar{r}_k 는 신호세기로부터 계산된 거리이며, 예측 상태 \bar{x}_{k+1}^- 는 사용자의 위치라고 예측한 좌표이다.

3.2 WLAN 기반 EKF 추적 알고리즘

제안하는 측위 알고리즘은 WLAN 기반의 EKF 방법이라는 특징과 측위와 더불어 추적까지 한다는 특징이 있다. 본 알고리즘은 <그림 1>의 단계 4, measurement update 단계에서 AP 신호의 RSSI(Received Signal Strength Indicator:

<그림 1> EKF 과정

을 선형화하면 $\delta X_{k+1} = \phi_k \delta X_k + W_k,$



<그림 2> 거리와 신호세기의 관계

수신신호세기 지표)로 거리를 측정함으로써 RSSI와 거리간의 관계식을 구해야 한다. 2.2.1에서 관계식을 소개하고, 2.2.2에서 알고리즘을 제안한다.

3.2.1 신호세기와 거리 간의 관계식

WLAN 환경에서 거리와 신호세기의 관계를 구하기 위해서 AP로부터 1m 거리마다 RSSI를 300회씩 측정 후, 측정치들의 평균을 구하여

```

algorithm TrackingEKF(AP_Positions, Distances)
// AP_Positions: Array of APs' positions.
// Distances: Array of estimated distances from the mobile terminal to APs.

// Step 1) Initial guess
1 initializes X_matrix and P_matrix
// Step 2) Linearizing.
2 for (i = 0; i < number of APs i++)
    2-1 Temp = square of (AP_Positions[i][0] - X_matrix[0])
        + square of (AP_Positions[i][1] - X_matrix[2])
        + square of (AP_Positions[i][2] - X_matrix[4]);
    2-2 r_0[i] = square root of Temp; // computed distance between nominal point and the i-th AP
3 for (i = 0; i < number of APs i++)
    3-1 for (j = 0; j < number of axes j++)
        3-1-1 if (((X_matrix[j*2] - AP_Positions[i][j]) is 0) or (r_0[i] is 0))
            then H_matrix[i, j*2] = 0
        3-1-2 else then H_matrix[i, j*2] = (X_matrix[j*2] - AP_Positions[i][j]) / r_0[i];
// Step 3) Kalman Gain
4 Temporary_K = H_matrix * P_matrix * H_matrix.Transpose() + R_matrix;
5 Inverse_K = Temporary_K.Inverse();
    // If K is not invertable then terminate
6 If(Inverse_K is not the inverse of K) then return false;
7 K_matrix = P_matrix * H_matrix.Transpose() * Inverse_K
// Step 4) Measurement update, Step 6) State propagation.
8 for (i = 0; i < number of APs i++)
    8-1 Z_matrix[i] = Distances[i] - r_0[i];
9 X_matrix = Pie_matrix * (X_matrix + K_matrix * Z_matrix)
// Step 5) Update error covariance, Step 6) State propagation
10 P_matrix = (I_matrix - K_matrix * H_matrix) * P_matrix
11 P_matrix = Pie_matrix * P_matrix * Pie_matrix.Transpose() + Q_matrix;
// Step 7) Goto step2
12 if(continue) then Update Distances and Goto Line 2
    else return true;

end TrackingEKF
    
```

<그림 3> EKF 추적 알고리즘

<그림 2>와 같이 도표를 작성한다. X-축은 AP와 측정 장소간의 거리이고, Y-축은 측정된 RSSI의 평균값이다. 실선은 RSSI와 거리의 회귀 분석 결과이며, 거리와 신호세기의 관계식도 <그림 2>에 표시되어 있다. <그림 2>의 관계식을 기본으로 사용자와 AP 사이에 존재하는 장애물을 감안하여 사용자의 위치에서 AP들까지의 거리를 구한다.

예를 들어, 현재 위치에서 판독한 RSSI가 -60이고 사용자와 AP 사이에 벽이 하나 존재한다면, 벽에 의하여 감쇄된 값 5를 판독 값에 더하여 -55를 관계식의 y 값으로 함으로써 다음 식을 얻는다.

$$-55 = -12\ln(x) - 33.66.$$

따라서 $x=5.918\text{m}$ 이다.

3.2.2 EKF 추적 알고리즘

본 연구에서 제안하는 추적 알고리즘은 <그림 3>과 같다. Line 1은 <그림 1>의 Step 1)의 과정으로 \bar{x}_0^- 행렬과 P_0^- 행렬을 초기화하는 단계로 본 연구에서는 <표 1>에 보이는 값으로 초기화한다. \bar{x}_0^- 의 0, 2, 4 항은 각각 사용자 위치의 x, y, z 좌표를 나타내고, 1, 3, 5항은 각각 x, y, z 좌표 방향으로 움직이는 사용자의 속도를 나타낸다. 초기 값으로 0을 배정하는 것은 사용자의 위치가 원점이라고 가정하는 것이다. 이 가정은 실제 위치와 오차가 크므로 P_0^- 행렬을 큰 값으로 초기화함으로써 빠르게 수렴할 수 있게 한다. 알고리즘의 X_matrix와 P_matrix는 이

<표 1> EKF 추적을 위한 인수들의 초기 값

Description	Value
\bar{x}_0^-	$[0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$
P_0^-	$\begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$
R_0	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Q_0	$\begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 \end{bmatrix}$

두 행렬에 대한 Matrix Class 객체이다.

이 알고리즘에서는 행렬 연산이 주로 사용됨으로, 이를 간편하게 하기 위하여 행렬의 요소를 Decimal Type(C# 데이터 유형)까지 계산할 수 있는 Matrix Class를 작성하였다. 이 Matrix Class에는 행렬의 합과 곱 등의 연산과 전치행렬, 역행렬을 구하는 메소드를 정의하였다. 구현한 역행렬 구하는 방법은 여인자전개법이다.

또한 H_matrix, K_matrix, I_matrix, R_matrix, Q_matrix, Pic_matrix는 <그림 1>에 사용된 H, K, I, R, Q, Φ 를 의미하는 Matrix Class의 객체이다. 그리고 r_0[number of APs]는 \vec{x}_0 에서 각 AP 들까지의 거리를 저장할 배열이며, Z_matrix는 <그림 1>의 Step 4) 중 $\vec{r}_k - \vec{r}_0$ 를 계산한 배열이다. 알고리즘의 Line 2~3은 r_0와 H_matrix를 구하는 과정이고, Line 4~7은 <그림 1>의 Step 3)의 Kalman gain을 구하는 과정이다. Line 8과 9는 각각 $\vec{r}_k - \vec{r}_0$ 와 $\hat{x}_{k+1} = \Phi_k(\hat{x}_k + K_k(\vec{r}_k - \vec{r}_0))$ 를 계산하며, Line 10과 11은 각각 $(P_k = (I - K_k H_k)P_k^-)$ 와 $P_{k+1} = \Phi_k P_k \Phi_k^T + Q_k$ 를 계산한다. Line 12는 <그림 1>의 Step 7)을 수행하는 것으로 다음과 같은 종료 조건이 거짓일 경우에는 Step 2)로 이동하여 위의 연산들을 반복

수행한다.

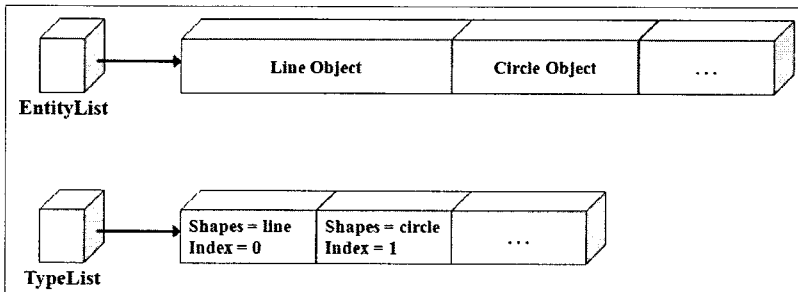
$$(k^{th} X_matrix - (k-1)^{th} X_matrix) \leq \text{Threshold or (number of iterations} \leq \text{limit)}$$

IV. 지도보기 프로그램

본 연구에서 구현하는 지도보기 프로그램은 Autocad로 작성한 DXF 유형(부록 A 참조)의 지도 파일을 입력받아 화면에 출력하고 조작하며 측위 결과를 표시한다. DXF 유형은 부록 A에 소개하고 여기서는 지도보기 모듈을 입력부분, 그리기 부분, 제어부분으로 나누어 제안한다.

4.1 입력부분

입력 부분은 DXF 파일을 도형의 유형(LINE, CIRCLE, ARC, ...) 단위로 읽어 해당 개체를 생성하여 엔터티 리스트에 추가한다. 이때 엔터티 리스트에 추가된 개체의 유형 이름을 인덱스와 함께 유형 리스트에 추가하여 둔다. 이렇게 하는 이유는 엔터티 즉 도형의 종류가 여러 가지 인데 모든 유형의 엔터티들을 하나의 리스트에 모두 기록함으로써 새로운 유형의 엔터티가 추



<그림 4> 부록 A, <표 3>의 입력 결과

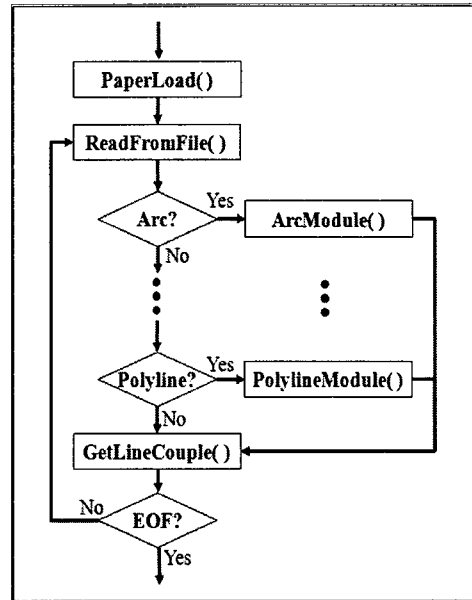
가로 만들어지더라도 본 소프트웨어를 그대로 사용할 수 있도록 하기 위함이다. 예를 들어 부록에 보이는 <표 3>의 DXF 파일을 입력하면 엔터티리스트(EntityList)와 유형리스트(TypeList)는 <그림 4>와 같이 작성된다.

DXF 파일로부터 도면정보를 읽어오는 과정을 순서도로 보이면 <그림 5>와 같다. DXFImporter Class에서 PaperLoad() 메소드가 호출되면 Dialog 창을 이용하여 DXF 파일의 경로를 구한다. 그 다음 ReadFromFile() 메소드가 호출되어 GetLineCouple() 메소드를 통해 DXF 파일에서 EOF까지 두 줄씩 데이터를 읽어온다. 부록 <표 3>의 DXF 파일 예에서 알 수 있는 바와 같이 현재 읽은 줄이 새로운 엔터티의 시작인지 아닌지는 읽어 온 두 줄 중 첫줄이 "0"이고 다음 줄이 "LINE", "CIRCLE", 등의 도형 유형 이름인지 보면 알 수 있다. 따라서 현재 읽은 줄이 엔터티 정보의 시작이면 해당 엔터티 정보를 읽는 메소드를 호출한다.

엔터티 정보를 읽는 각 메소드는 해당 도형을 결정하는 요소들을 입력받아 엔터티 개체를 생성하여 엔터티 리스트에 추가한다. <그림 5>에 보이는 바와 같이, 이때 엔터티 유형과 해당 개체의 엔터티 리스트에서의 위치를 나타내는 인덱스를 유형리스트에도 추가한다.

4.2 그리기 부분

그리기 메소드는 유형리스트를 구성하는 개체들을 하나씩 차례로 접근하여, 이 개체의 인덱스 필드의 값으로 이 개체와 연관된 엔터티 리스트의 개체를 사용자 인터페이스인 윈도우에 그림으로써 전체 지도를 화면에 그린다. 각



<그림 5> DXF 파일을 읽어오는 과정의 순서도

엔터티 개체(예를 들면 Line 개체)에는 그리기 메소드가 멤버메소드로 정의되어 있으며, 도형을 결정하는 값들뿐만 아니라 지도의 중앙점 좌표 (paperMidX와 paperMidY), 윈도우의 중앙점 좌표 (windowX, windowY), 스케일, 등도 인수로 받는다. 각 인수들의 값을 구하는 방법은 다음과 같다.

지도의 중앙점 좌표는 출력될 지도의 중앙점으로, 지도 좌표들 중 X좌표의 최대값과 최소값의 차이의 반을 최소값에 더한 값이 지도 중앙점의 X좌표이며 중앙점의 Y좌표도 같은 방법으로 구한다. 예를 들어, <예 1>의 <그림 6>과 같이 지도를 구성하는 도형이 하나의 직선뿐이며, 이 직선의 시작점의 좌표가 (600, 500)이고 끝점의 좌표가 (10, 20)인 경우에, 지도의 중앙점의 좌표는 (305.0, 260.0)이며 이 값을 구하는 식은

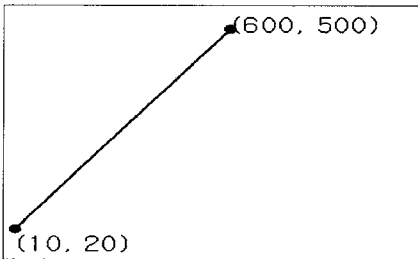
<예 1>의 paperMidX와 paperMidY에 보인다. 일반적으로 지도의 중앙점을 구하는 식은 다음과 같다:

$$\text{paperMidX} = \text{XMin} + (\text{XMax} - \text{XMin}) / 2;$$

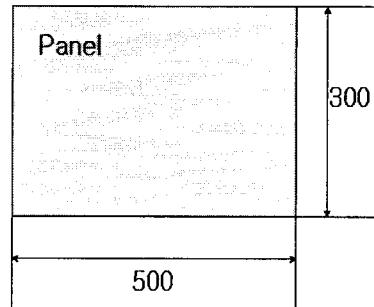
$$\text{paperMidY} = \text{YMin} + (\text{YMax} - \text{YMin}) / 2;$$

지도의 중앙점은 사용자가 지도를 상하좌우

<예 1> 지도중앙점, 윈도우 중앙점, 스케일 등을 구하여 그리기



<그림 6> 직선 하나로 이루어진 지도의 예



<그림 7> 지도가 출력될 화면의 예

<그림 6>의 지도를 <그림 7>의 화면에 출력할 때, 최대 X, 최소 X, 최대 Y, 최소 Y는 각각 다음과 같이 된다.

$$\begin{aligned} \text{XMax} &= 600.0 \\ \text{XMin} &= 10.0 \\ \text{YMax} &= 500.0 \\ \text{YMin} &= 20.0 \end{aligned}$$

지도의 중앙점은 다음과 같다.

$$\begin{aligned} \text{paperMidX} &= 10.0 + (600.0 - 10.0) / 2.0 = 305.0 \\ \text{paperMidY} &= 20.0 + (500.0 - 20.0) / 2.0 = 260.0 \end{aligned}$$

화면의 중앙점은 다음과 같다.

$$\text{windowMid} = (250.0, 150.0)$$

스케일은 다음과 같이 구한다.

$$\begin{aligned} \text{scaleX} &= 500.0 / (600.0 - 10.0) = 0.847\dots \\ \text{scaleY} &= 300.0 / (500.0 - 20.0) = 0.625 \end{aligned}$$

$$\text{mainScale} = 0.625 * 0.9 = 0.5625$$

다음과 같이 좌표를 변환하여 그리면 <그림 8>의 결과를 얻는다.

$$\begin{aligned} \text{시작X} &= (600.0 - 305.0) * 0.5625 + 250.0 = 415.9375 \\ \text{시작Y} &= -(500.0 - 260.0) * 0.5625 + 150.0 = 15.0 \\ \text{끝X} &= (10.0 - 305.0) * 0.5625 + 250.0 = 84.0625 \\ \text{끝Y} &= -(20.0 - 260.0) * 0.5625 + 150.0 = 285.0 \end{aligned}$$

로 움직일 때 변하게 된다.

윈도우의 중앙점 좌표는 지도가 출력될 영역의 중앙점으로 다음 식으로 구한다. <예 1>의 <그림 7>과 같은 경우에는 좌표 (250.0, 150.0)이 윈도우의 중앙점이다.

$$\text{windowMidX} = \text{drawPanelWidth} / 2;$$

$$\text{windowMidY} = \text{drawPanelHeight} / 2;$$

지도 좌표를 윈도우 좌표로 변환하기 위하여 지도 영역의 크기에 대한 윈도우 영역의 크기의 비율, 즉 스케일을 구한다. 예를 들어 <예 1>의 경우, x 축의 스케일은 $500 / (600 - 10)$ 이고, y 축의 스케일은 $300 / (500 - 20)$ 이다. x 스케일과 y 스케일 중, 작은 것을 적용해야 지도 전체를 윈도우에 나타낼 수 있다. 또한 이 스케일을 그대로 적용하면 지도가 화면에 꽉 들어참으로 90%로 줄인 스케일을 그리기에 적용한다. <예 1>에는 mainScale에 그리기에 적용하는 스케일을 계산하는 과정이 보인다.

그리기는 입력에서 작성한 엔터티 리스트를 구성하는 엔터티들을 하나씩 차례로 화면에 출력하는데, 이때 지도좌표 (paperX, paperY)를 비롯한 엔터티를 결정하는 요소들을 지도의 중앙

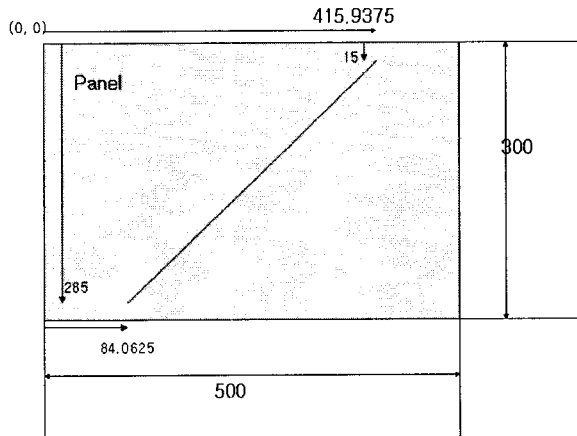
점 (paperMidX와 Y), 윈도우의 중앙점 (windowMidX와 Y) 그리고 스케일을 사용하여 윈도우 좌표 (winX, winY)로 변환한다. 좌표 변환 방법은 주어진 지도 좌표와 지도의 중앙점 간의 거리에, 스케일을 곱한 결과를 화면의 중앙점 좌표에 더한다. 이 방법을 식으로 표현하면 다음과 같다. 이때 윈도우의 Y좌표는 지도의 Y좌표와 반대로 위에서 아래 방향으로 증가하기 때문에 음수를 취해준다. <예 1>의 지도를 화면에 출력한 결과가 <그림 8>에 보인다.

$$\text{winX} = (\text{paperX} - \text{paperMidX}) * \text{mainScale} + \text{windowMidX};$$

$$\text{winY} = -(\text{paperY} - \text{paperMidY}) * \text{mainScale} + \text{windowMidY};$$

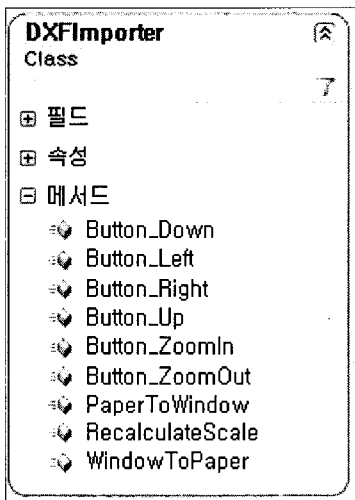
4.3 제어부분

화면에 출력된 지도를 조작하기 위한 사용자 인터페이스로 <그림 9>에 보이는 바와 같은 ZoomIn, ZoomOut, Up, Down, 등의 버튼을 제공한다. Button_ZoomIn() 메소드가 호출되면 현재



<그림 8>스케일을 적용한 그리기 결과

Scale을 증가($mainScale \times 1.2$)시켜 도면을 확대 하며, Button_ZoomOut() 메소드가 호출되면 현재 Scale을 감소($mainScale \div 1.2$)시켜 도면을 축소한다. 이후 최초의 도면크기를 원할 땐 RecalculateScale() 메소드를 호출하면 된다.



<그림 9> DXFImporter Class의 Control Part

Button_Up(), Button_Down(), Button_Left(), Button_Right() 메소드는 도면을 상하좌우로 이동하는 메소드이며, PaperToWindow() 메소드는 도면의 좌표를 DrawPanel 좌표로 변환하고, WindowToPaper() 메소드는 DrawPanel의 좌표를 도면의 좌표로 변환해 주는 메소드이다. <표 2>에서는 각 메소드에 대해 계산하는 방법을 나타내고 있다.

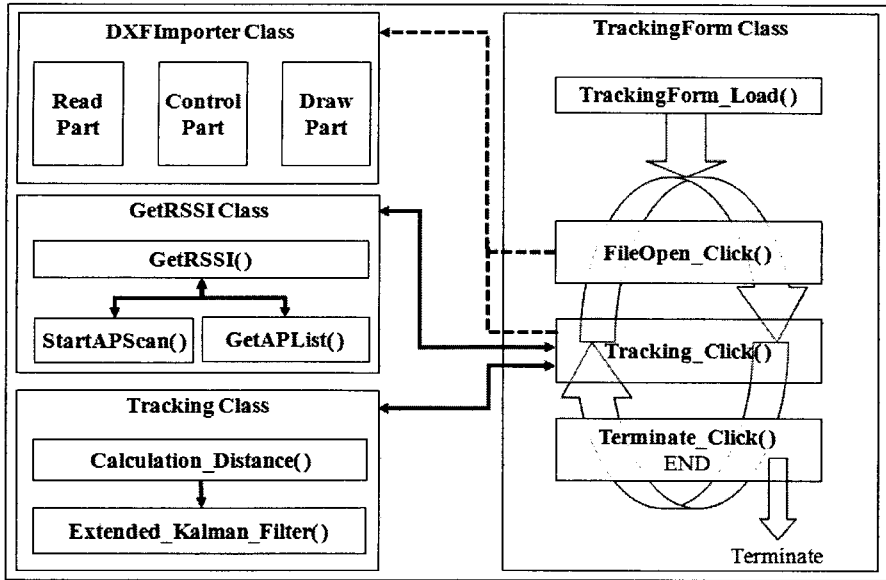
V. 통합 시스템 구현 및 실험

5.1 통합 시스템 구현

통합 시스템은 Intel(R) PRO/Wireless 2200BG Network Connection 랜 카드가 장착된 Lap-Top 컴퓨터에서 Microsoft Visual Studio .NET 2005의 C# 언어를 사용하여 구현한다. 지도보기 모듈과 측위 모듈을 <그림 10>과 같이 통합하였다.

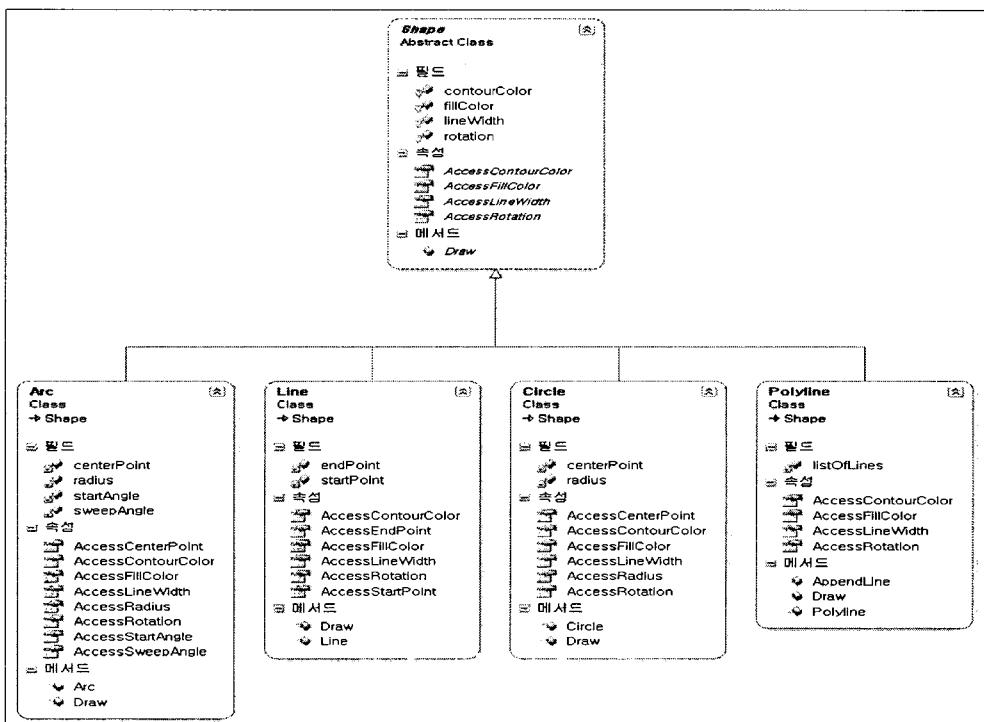
<표 2> Control Part의 메소드

메소드	계산 방법
Button_ZoomIn()	$mainScale = mainScale * 1.2;$
Button_ZoomOut()	$mainScale = mainScale / 1.2;$
Button_Up()	$paperMidY = paperMidY + 30 / mainScale;$
Button_Down()	$paperMidY = paperMidY - 30 / mainScale;$
Button_Left()	$paperMidX = paperMidX - 30 / mainScale;$
Button_Right()	$paperMidX = paperMidX + 30 / mainScale;$
Window ToPaper()	$paperX = (winX + paperMidX * mainScale - windowMidX) / mainScale;$ $paperY = (-winY + paperMidY * mainScale + windowMidY) / mainScale;$
Paper ToWindow()	$winX = (paperX - paperMidX) * mainScale + windowMidX;$ $winY = -(paperY - paperMidY) * mainScale + windowMidY;$



<그림 10> 통합 시스템의 이벤트 흐름도

FileOpen_Click() 이벤트가 발생되면 DXFImporter Class에서 DXF파일에 저장된 실내 도면정보를



<그림 11> 도형 유형의 Class와 추상 Class에 대한 상속 관계

읽어와 DrawPanel에 출력하게 된다. 여기서 DXFImporter Class를 크게 DXF 파일을 읽어오는 Read Part와 읽어온 도면정보를 제어하는 Control Part, 그리고 도면정보를 출력하는 Draw Part로 나눌 수 있다. 도면정보의 요소로는 Arc, Circle, Line, Polyline이 사용되었으며, 각 모양에 대한 Class는 <그림 11>처럼 추상 Class인 Shape Class로부터 상속받아 Interface를 재사용한다.

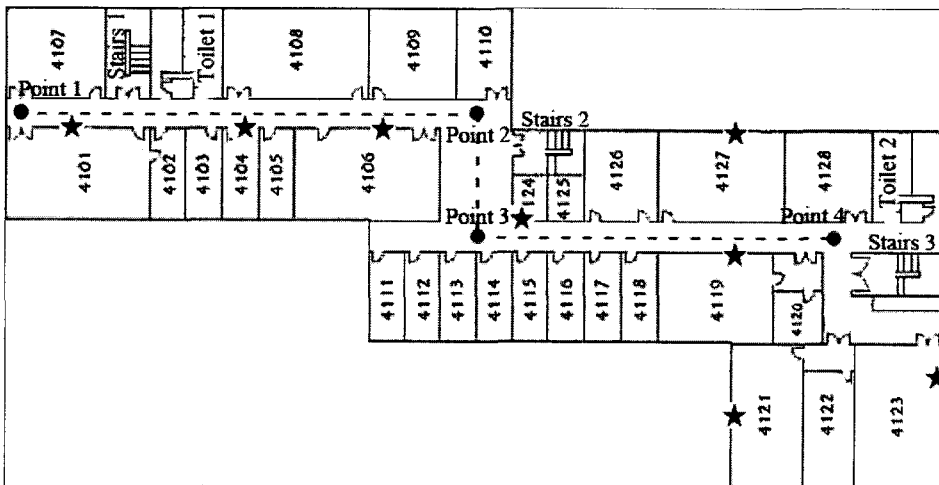
이후 Tracking_Click() 이벤트가 발생되면 GetRSSI Class를 이용하여 현재 위치에서 AP들 각각의 RSSI를 판독한다. 그 다음 Tracking Class에서 Calculation_Distance() 메소드로 AP까지의 거리를 추정(propagation model 이용)하고 Extended_Kalman_Filter() 메소드로 사용자의 위치를 추적하게 된다. 계산된 사용자의 위치는 DXFImporter Class의 Draw Part를 이용하여 DrawPanel에 그려지게 되며, 그리진 도면과 사용자의 위치에 대해서 확대, 축소 및 이동에 대한 제어는 Control Part가 수행한다.

5.2 실험

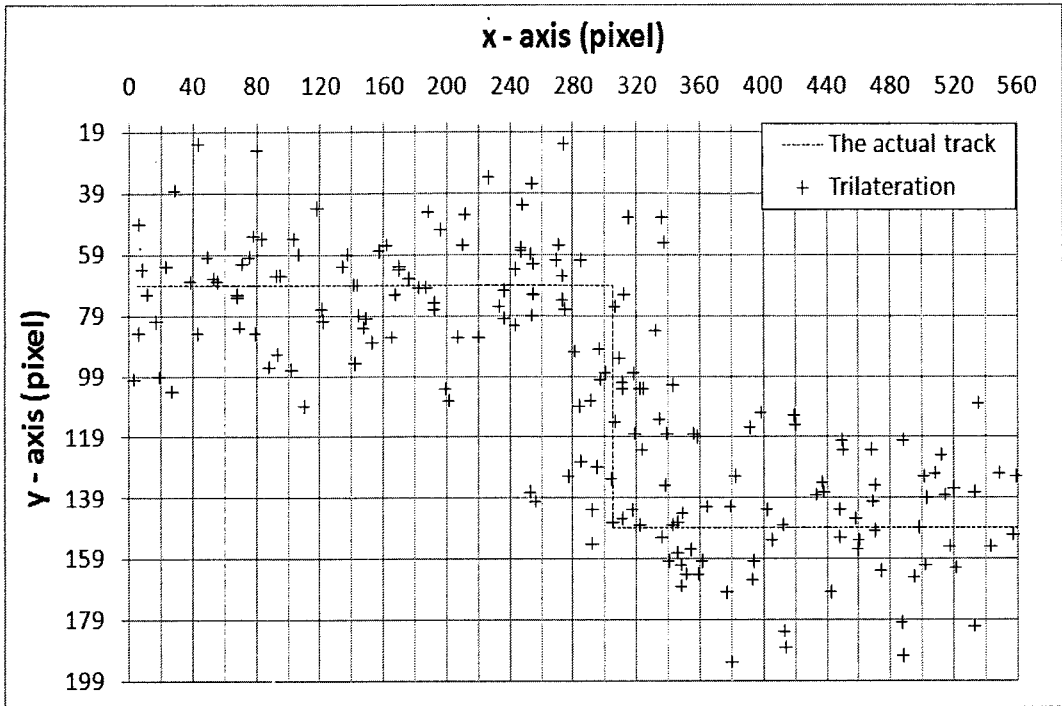
구현한 시스템을 사용하여 <그림 12>에 보이는 WLAN 환경의 자연과학관 4층(85.5m x 43.8m) 복도에서 사용자 추적 실험을 수행하였다. 실험에 사용된 AP들은 총 8개로 AP1부터 AP8까지 <그림 12>에 ‘★’로 표시된 곳에 위치한다.

복도의 왼쪽 끝에서 오른쪽 끝으로 걸어가면서, 우선 Lassabe(2005)의 방법을 적용하여 이동 단말기의 궤적을 측정하여 보았다. 실험 결과는 <그림 13>에 + 기호로 나타냈으며, 측위 결과에 이동단말기의 이동 패턴이 반영되지 않고 오차 평균 범위에서 무작위적임을 알 수 있다. 오차 평균은 3.45m이고 표준편차는 1.78m이다.

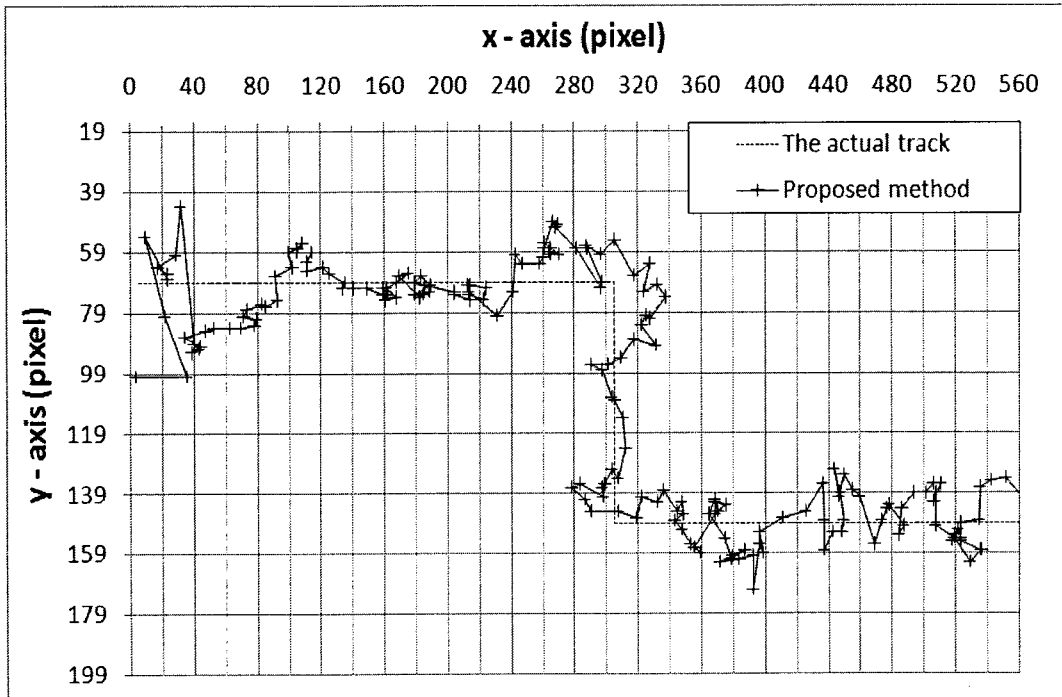
똑 같은 실험에서 이번에는 3절에서 소개한 EKF 방법을 적용하였다. 실험결과는 <그림 14>에 보인다. 실험 초기와 진행 방향이 바뀌는 부분에서는 오차가 크지만 진행 방향이 일정한 부분에서는 실제 궤적이 가까운 추적 결과를 보임을 알 수 있다. 이때의 오차평균은 1.81 m이고 표준편차는 0.91m이다.



<그림 12> 실험 환경



<그림 13> 기존의 방법을 적용한 실험 결과

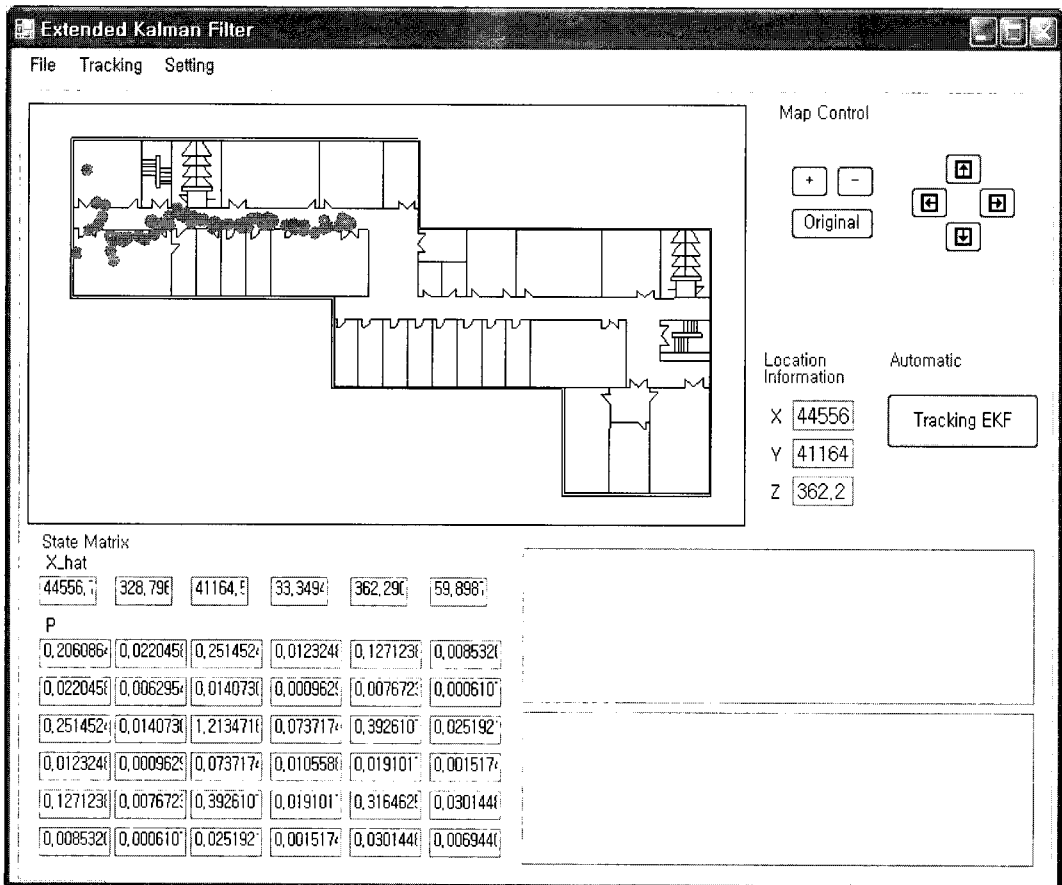


<그림 14> 제시하는 EKF 방법을 적용한 실험 결과

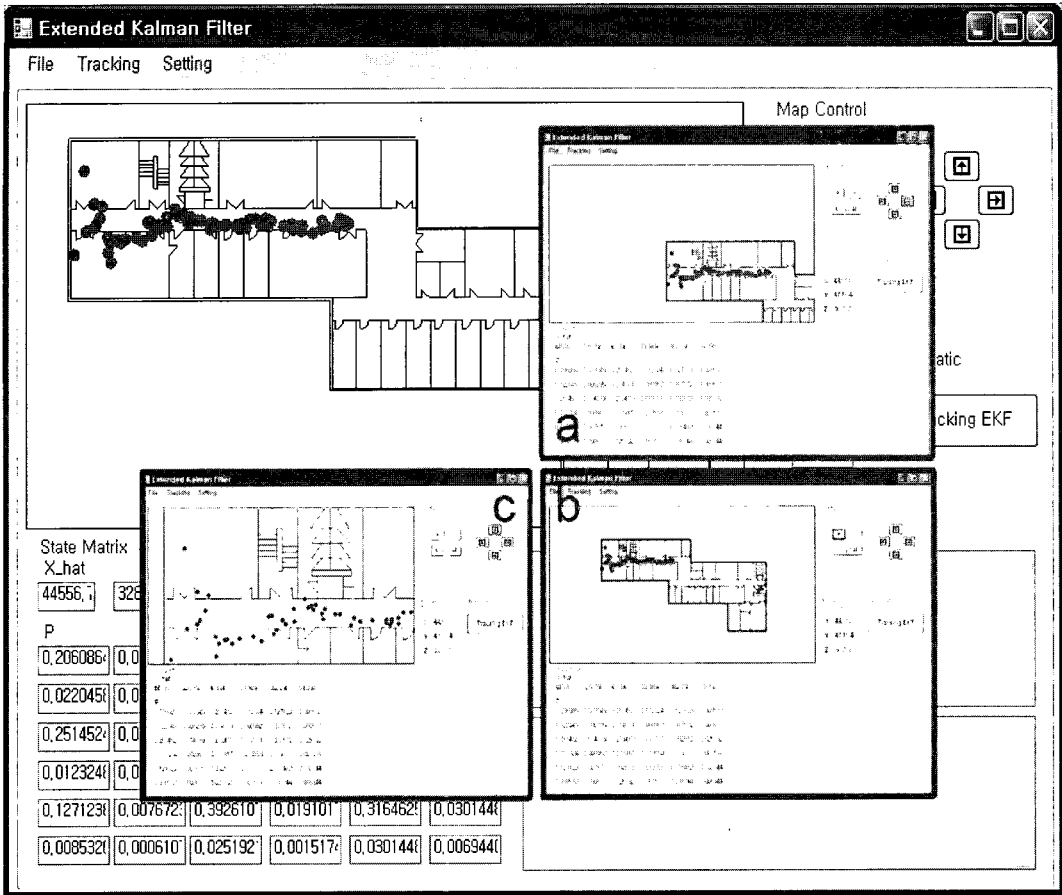
다음은 지도보기 모듈을 실험하였다. 본 실험에서는 북도의 Point 1지점에서 Point 2지점까지 이동하면서 사용자의 위치를 판정하여 도면정보가 출력된 DrawPanel에 위치를 출력하며, 사용자의 이동 경로와 도면에 대해서 이동 및 확대, 축소 기능을 실험한다.

통합 시스템의 실행 화면은 <그림 15>에 나타내 보였다. DXF 지도 파일이 좌상단 DrawPanel에 출력되고 EKF를 이용한 사용자 추적 결과가 지도상에 표시된 것을 볼 수 있다. 실행 화면의 Map Control 그룹박스의 제어 버튼을 이용하여 도면을 이동, 확대 및 축소할 결

과가 <그림 16>에 나타나 있다. Map Control 그룹박스에서 화살표가 그려진 상하좌우 버튼을 누르면 DrawPanel 안에서 도면이 이동하는 것을 볼 수 있으며, '+'가 그려진 확대 버튼을 누르면 추적 경로를 자세히 볼 수 있는 반면 도면의 중심을 기준으로 지도가 확대되기 때문에 사용자의 위치가 DrawPanel에서 사라질 수도 있다. 이 경우엔 다시 이동 버튼을 사용하여 도면의 원하는 지역이 중심이 되도록 이동할 수 있다. 또한 '-'가 그려진 축소 버튼을 누르면 도면이 축소되어 사용자가 자연과학관 4층 중 어느 부근에 있는지를 한 눈에 알아 볼 수 있다.



<그림 15> 통합시스템의 사용자 인터페이스



<그림 16> Control Part의 도면 이동 및 확대, 축소 결과 a: 이동, b: 축소, c: 확대

VI. 결론

위치기반서비스는 사용자에게 매우 유용한 정보를 제공함으로써, 옥외에서 뿐만 아니라 옥내에서도 제공되어야 한다. 그러나 옥외용 위치기반서비스는 상용화가 활발히 이루어지고 있는데 반하여 옥내용 위치기반서비스의 상용화에는 매우 희박하다. 일반적으로 위치기반서비스 시스템을 구성하는 여러 모듈 가운데에는 측위모듈과 지도조작 모듈이 반드시 포함된다. 본 연구에서는 옥내측위 모듈과 지도조작 모듈을 통합하여 개발한 시스템의 결과를 제시하였다.

본 연구에서 개발한 옥내측위 모듈은 무선근거리통신망(WLAN) 환경에서 사용자의 위치를 판정함으로써 특수 장비를 요구하지 않는다는 경제적인 장점을 가지고 있다. 또한 본 연구의 옥내측위에 적용한 기술은 EKF 방법으로 기존의 EKF 방법이 블루투스 환경의 옥내측위에 그친 반면, 본 연구에서 개발한 EKF 방법은 WLAN 환경에서 위치추적까지 한다는 점에서 기존의 방법과 크게 다르다.

본 연구에서 개발한 지도조작 모듈은 Autocad로 작성한 DXF 형식의 옥내 지도 파일을 입력받아 사용자 인터페이스에 출력하고, 출력한 지

도를 확대·축소·이동하는 기능을 제공한다. 또한 옥내 측위 모듈이 제공하는 사용자의 현재 위치를 동적으로 지도상에 표시하는 기능도 제공한다. WLAN 환경에서 EKF 방식을 적용한 위치추적 방식을 지도보기 모듈로 구현함으로써 본 연구 결과가 상용 위치기반서비스를 위한 시스템 개발에 쉽게 활용될 수 있을 것으로 사료된다.

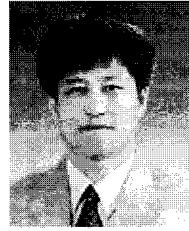
향후에는 옥내 위치기반서비스에 유용한 지도정보를 분석하여 지도 파일에 이러한 정보를 삽입하고, 또한 지도정보를 이용하여 측위 및 추적 모듈의 정확도를 제고하는 방안을 연구할 필요가 있다. 나아가서 옥내 위치기반서비스 시스템의 상용화를 위한 요소 기술들도 계속 개발할 필요가 있다.

참고문헌

- 김태성, 전효정, 남광우, “위치기반서비스의 비즈니스 모델,” 한국통신학회논문지, 제31권, 제9B호, 2006, pp. 848-856.
- 박종현, 김문구, 백종현, “위치기반서비스(LBS)의 산업구조 분석 및 시장개발전략 방향,” 한국통신학회지(정보통신), 제20권, 제4호, 2003, pp. 478-488.
- 임재걸, 주재훈, 남윤석, “위치기반서비스를 위한 통합측위시스템 설계 및 응용,” *Journal of Information Technology Applications & Management*, Vol. 13, No. 4, 2006, pp. 57-70.
- 주재순, 염경환, 이상정, “GPS/ GLONASS 통합 수신용 RF 전단부의 설계 및 제작,” 정보과학회논문지, 제12권, 4호, 2001, pp. 1226-2285.
- Bahl, P. and Padmanabhan, V., “RADAR: An in-building RF-based user location and tracking system”, *INFOCOM 2000*, Mar. 2000, pp. 775-784.
- Bruntrup, R., Edelkamp, S., Jabbar, S., and Scholz, B., “Incremental Map Generation with GPS Traces,” *Proceedings of IEEE Intelligent Transportation Systems*, 2005. (13-15 Sept. 2005) pp. 574-579.
- Finkelstein, E., *AutoCAD 2007 and AutoCAD LT 2007*, Wiley Publishing, Inc., 2007.
- Gehrig, S., Wagner, S., and Franke, U., “System Architecture for an Intersection Assistant Fusing Image, Map, and GPS Information,” *Proceedings of IEEE Intelligent Vehicles Symposium*, 2003 (9-11 June 2003), pp. 144-149.
- Gelb, A., *Applied Optimal Estimation*, The M.I.T. Press, 1974.
- Han, M. and Xue, T., “Design and Realization of GPS and GIS Integration Technology in Wetland Study,” *Proceedings of 2004 IEEE International Geoscience and Remote Sensing Symposium*, Vol. 5, pp. 2857-2860.
- Harter, A. and Hopper, A., “A New Location Technique for the Active Office,” *IEEE Personal Communications* 4, 5 (Oct. 1997), pp. 43-47.
- Hwang, P. and Brown, R. G., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons., 1992.

- Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *T-ASME*, 1960, pp. 35-45.
- Kotanen, A., Hannikainen, M., Leppakoski, H., and Hamalainen, T. D., "Experiments on Local Positioning with Bluetooth", *Proceedings of the International Conference on Information Technology: Computers and Communications, IEEE*, April 2003, pp. 297-303
- Lassabe, F., Canalda, P., Chatonnay, P., and Spies, F., "A Friis-Based Calibrated Model for WiFi Terminals Positioning," *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, 2005, (13-16 June 2005) pp. 382-387.
- Maybeck, P. S., *Stochastic Models, Estimation, and Control Vol. I, II and III*, Academic Press, 1982.
- Priyanthat, N., Chakraborty, A. and Balakrishnan, H., "The Cricket Location-Support System," Proc. of 6th ACM International Conference on Mobile Computing and Networking, Boston, MA, Aug. 2000.
- Want, R., Hopper, A., Falcao, V., and Gibbons, J., "The Active Badge Location System," *ACM Transactions on Information Systems* 10, 1 (January 1992), pp. 91-102.
- Yang, D., Cai, B., and Yifang, Y., "An Improved Map-Matching Algorithm Used in Vehicle Navigation System," *Proceedings of 2003 IEEE Intelligent Transportation Systems*, 2003, Vol. 2, (12-15 Oct. 2003) pp. 1246-1250.

임재걸 (Yim, Jaegeol)



현재 동국대학교(경주캠퍼스) 공학대학 컴퓨터멀티미디어학과 교수로 재직 중이다. 일리노이 주립대학교에서 컴퓨터과학 박사학위를 취득하였다. 주요 관심 분야는 페트리 넷 이론과 응용, 시스템 분석, 컴퓨터 네트워크, 인공지능 등이다. 한국정보과학회논문지, 한국정보처리학회논문지, 대한전자공학회논문지, 멀티미디어학회논문지, Information Sciences 등에 다수의 논문을 게재하였다.

주재훈 (Joo, Jaehun)



현재 동국대학교(경주캠퍼스) 경영·관광대학 전자상거래학과 교수로 재직 중이다. 부산대학교에서 경영학박사 학위를 취득하였다. 주요 관심분야는 전자상거래와 e-비즈니스 전략/모형/개발방법론, 시맨틱 웹, 상황인지 및 유비쿼터스 응용, 웹기반 DSS 등이다. 경영학연구, 경영정보학연구, 한국경영과학회지, Information Systems Management, International Journal of Industrial Engineering, Expert Systems with Applications, Journal of Computer Information Systems 등에 다수의 논문을 게재하였다.

정승환 (Jeong, SeungHwan)



현재 동국대학교 대학원 전자계산학과 석사과정 재학 중이다. 동국대학교에서 공학대학 컴퓨터학과 공학사를 취득하였다. 주요 관심 분야는 LBS, GIS, 802.11 표준 등이다. 한국정보과학회, 한국정보처리학회의 학술발표 논문집에 다수의 논문을 게재하였고, JCIT, 멀티미디어학회 논문지에 논문을 게재하였다.

부록 A: DXF 파일 형식

DXF 파일 형식은 Autocad 파일 형식 중 하나로, 다른 소프트웨어에서 읽기에 편리하다는 특징이 있다. 예를 들어, Autocad로 작성한 도면이나 지도를 DXF 형식으로 저장하면 일반적인 텍스트 파일로 저장됨으로 메모장이나 다른 소프트웨어로 파일 내용을 읽어 볼 수 있다.

DXF 파일 형식과 관련하여 본 논문에서 필요로 하는 사항들은 다음과 같다. DXF 파일에서는 정보가 그룹코드와 쌍을 이루어 저장된다. 이때 그룹코드는 정보의 유형을 지정하는 역할을 한다. 정보의 유형에는 텍스트, 정수, x좌표, 등 여러 가지가 있다. 예를 들어 <표 3>에서 그룹코드 0은 어떤 정보 단위의 시작이나 끝임을 나타내며 쌍을 이루는 항은 어떤 유형의 정보단

<표 3> DXF 파일의 예

그룹코드	의 미
0 LINE	엔티티의 종류 line
8 0	이 엔티티가 속한 레이어명은 "0"
10 35.88862818985399	라인의 시작 x 좌표 값
20 13.65995672501049	라인의 시작 y좌표 값
30 0.0	라인의 시작 z좌표 값
11 58.15630849450378	라인의 끝 x좌표 값
21 40.68115047870072	라인의 끝 y좌표 값
31 0.0	라인의 끝 z좌표 값
0 CIRCLE	엔티티 종류 circle
8 0	이 엔티티가 속한 레이어명은 "0"
10 46.20050706958136	중심점 x 좌표
20 32.17171374022175	중심점 y 좌표
40 6.2544244956812	반지름의 길이
...	...

<표 4> 엔터티 유형에 따른 그룹 코드와 의미의 예

Entity Type	그룹 코드	의 미
ARC	10	중심점의 X 좌표
	20	중심점의 Y 좌표
	40	반지름
	50	시작 각
	51	끝 각
CIRCLE	10	중심점의 X 좌표
	20	중심점의 Y 좌표
	40	반지름
LINE	10	시작점의 X 좌표
	20	시작점의 Y 좌표
	11	끝점의 X 좌표
	21	끝점의 Y 좌표
LWPOLYLINE	10	X 좌표
	20	Y 좌표
	70	다음 점과의 연결 상태 (연결=1, 비 연결=0)
	90	꼭지점의 수

위인지를 나타낸다. 즉, <표 3>의 두 번째 줄의 경우에는 LINE이라는 정보 단위의 시작임을 나타낸다. LINE 정보를 나타내는 영역에서, 그룹 코드 10은 쌍을 이루는 향이 선의 시작점의 x좌표, 20은 y좌표, 30은 z좌표임을 의미하고, 그룹 코드 11은 쌍을 이루는 향이 선의 끝점의 x좌표, 21은 y좌표, 31은 z좌표임을 의미한다.

엔터티의 종류에 따라 엔터티 정보를 구성하는 요소가 각각 다르다. LINE의 경우에는 시작점의 좌표와 끝점의 좌표로 엔터티 정보가 구성되며, 원의 경우에는 중심점의 좌표와 반지름으로 구성된다. 한편 호는 중심점의 좌표, 반지름의 길이, 호의 시작 각 그리고 끝 각으로 구성된

다. 본 연구에서 다루는 엔터티 유형과 각 엔터티를 구성하는 항목들을 요약하면 <표 4>와 같다.

<Abstract>

Integrated Indoor Positioning Systems Reflecting Map Information for Location Based Services

Jaegool Yim · Jaehun Joo · Seunghwan Jeong

So many location based service systems, including automobile navigation system, logistic management, taxi fleet management, and so on, are being used everywhere. However, these are all outdoors. This paper provides a stepping stone for commercial indoor location based services by developing an integrated system of our indoor positioning and map viewer modules. For the indoor positioning, we propose WLAN (Wireless Local Area Network) based EKF (Extended Kalman Filter) which estimates user's current location and tracks user's trace in the sequence of time. Our map viewer renders a map recorded in an Autocad DXF file and provides functions of map manipulation such as zoom-in, zoom-out, and move. We integrate our indoor positioning and map viewer modules and discuss the experimental results of the integrated system.

Keywords: Location Based Service, Indoor Positioning, Extended Kalman Filter, Location Tracking, Integrated Systems