

옥내 이동객체 데이터베이스의 갱신 전략 제안

임재걸*, 권기용*, 이강재**

*동국대학교(경주) 컴퓨터멀티미디어학과

**수원과학대학 컴퓨터정보과

e-mail : yim@dongguk.ac.kr

powergky@dongguk.ac.kr

kjlee@ssc.ac.kr

Introducing a New Strategy for Updating Indoor Moving Object Database

Jaegeol Yim*, Kiyong Gwon*, Kangjai Lee**

*Dept of Computer Multimedia, Dongguk University(geongju)

**Dept of Computer Information, Suwon Science College

요 약

본 논문은 옥내 이동객체 데이터베이스의 갱신 전략을 제안한다. 제안하는 방법은 최근의 일련의 측정치에 칼만 필터를 적용하여 이동객체의 속도를 예측하고, 예측한 속도를 최근 측정치에 적용하여 현재 위치를 추정한다. 그리고 현재 위치로 추정된 값과 현재 실측값의 차이가 일정 거리 이하이면 실측값을 전송하지 않는다. 제안하는 방법의 효율성을 검증하기 위하여 실험 공간을 실제로 걸어가면서 측정된 일련의 측정치에 제안하는 방안을 적용하여 전송을 절약하는 비율과 이로 인한 오차의 증가율을 분석한다.

키워드: 이동객체 데이터베이스(Moving Object Database), 칼만 필터(Kalman Filter), 옥내 측위(Indoor Positioning)

I. 서 론

본 논문은 옥내 위치기반서비스를 위한 이동 객체 데이터베이스 시스템의 갱신 방안을 제안한다. 옥내 이동 객체는 옥내를 보행하는 사람이 휴대한 초소형 노트북컴퓨터나 PDA 혹은 스마트폰 등의 이동단말기를 지칭한다. 옥내 이동물체 데이터베이스의 목적은 이동단말기의 위치를 시간 변화에 따라 추적함으로써 "누가 언제 어디에 있었는지"와 같은 시간과 공간에 대한 정보를 제공하는 것이다.

시간과 공간에 대한 정보는 위치기반서비스에 반드시 필요한 요소임으로 자동차 관리, 택배, 운송, 군사 등의 분야에서 응용소프트웨어 개발을 위한 시공간 정보 기술이 널리 연구되었다. 그러나 지금까지 개발된 시공간 정보 기술은 모두 옥외용이고 위치 판단은 주로 GPS에 의존한다. 그런데 GPS 신호는 옥내에서 잡히지가 않기 때문에 옥외용으로 개발된 기존의 기술들은 옥내에서 무용지물이 된다.

그럼에도 불구하고 시간과 공간 정보는 옥내에서도 매우 중

요하다. 예를 들어, 코엑스와 같은 대형 전시장에는 수많은 단체 관람객들이 내방한다. 단체 관람은 많은 사람들이 정해진 스케줄에 따라 움직여야 하기 때문에 인솔자는 항상 회원들의 위치에 신경을 쓴다. 특히 초등학생의 경우에는 더욱 그렇다. 나아가서, 학생 단체 관람의 경우에는, 선생님이 교육적 효과를 평가하기 위하여 얼마나 많은 학생이 어느 부스를 얼마나 오래 동안 방문하였는지 통계적 자료를 원하기도 한다. 이러한 통계적 자료는 전시회 주관기관이 전시물들에 대한 관람객의 관심도를 알아보는 데 필요하기도 하다.

옥내 위치기반서비스의 또 다른 예로 박물관 관람 가이드 시스템을 들 수 있다. 박물관 관람 가이드 시스템은 관람객이 소지하는 소형 이동 단말기로 사용자의 위치와 전시물의 위치를 파악하여 사용자가 현재 보고 있는 전시물과 관련된 콘텐츠를 제공하거나 관람객에게 이동 경로를 제안하는 등 유용한 서비스를 제공하는 시스템이다. 이러한 시스템은 실감나는 현장 학습과 같은 효과를 나타낸다.

옥내위치기반 서비스 시스템 개발에 필요한 요소 기술에는

옥내 이동객체의 위치를 파악하는 측위 기술과 이동객체의 이력을 관리하고 질의에 응답하여주는 이동객체 데이터베이스 시스템 개발에 필요한 기술이다. 옥내 측위 기술에 대한 연구는 이미 상당한 수준에 이르고 있다. 옥내 위치기반서비스의 초기 연구는 [1-3]에 소개 되어 있고, UDP(User Datagram Protocol) 신호의 세기에 K-NN(K Nearest Neighbors) 방법을 적용한 옥내측위 방법[4]에 소개 된다. 이러한 방법들은 측위를 위한 특수 장치를 설비해야 하는 단점이 있어, 이를 극복하는 방법으로 무선근거리통신망 환경의 옥내 측위 방법이 [5-7]에 소개되었다. 또한 무선근거리 통신망 환경의 옥내 측위 방법의 효율성을 개선하는 방법으로 칼만 필터와 의사결정트리를 이용하는 방법이 각각 [8]과 [9]에 소개되었다.

옥내 이동객체데이터베이스 시스템의 효율성을 결정하는 요인들 중에는 이동 객체의 위치 정보를 갱신하는 정책이 있다. 위치정보 갱신은 이동객체가 자신의 위치를 서버에게 무선 통신으로 알려줌으로써 이루어지는데, 이동 객체의 수가 적은 경우에는 주기적인 갱신 정책도 가능하지만 이동 객체의 수가 매우 클 경우에는 통신량을 줄이면서 데이터의 정확도도 유지하는 정책을 사용하지 않으면 과도한 통신비용을 지출하게 된다. 따라서 본 논문은 옥내 이동객체 데이터베이스를 위한 갱신 정책을 제안한다.

제안하는 갱신 정책의 요지는 시점 t_0, t_1, \dots, t_i 각각에서 실측한 위치 정보에 칼만 필터를 적용하여 t_{i+1} 이후의 단말기의 위치를 추정하고, 추정된 위치와 실측 위치간의 차이가 일정 수준 이하이면 통신을 생략하는 전략을 쓰는 것이다.

II. 관련 연구

이동객체 데이터베이스 기술은 이미 심도 있게 연구되었다. 그러나 기존의 연구는 GPS 기반의 옥외용이고, 옥내 이동객체 데이터베이스 관련 연구는 아직 발표된 바가 전무하다. 따라서 본 절에서는 옥외용으로 이미 개발된 이동객체 데이터베이스 관련 기술을 소개한다. 또한 본 논문에서는 칼만 필터를 이용하여 이동객체의 위치를 추정하는 방법을 이용하려고 함으로 본 절에서는 칼만 필터도 소개한다.

1. 이동객체 데이터베이스

컴퓨터가 작아지고 무선통신망과 센서 기술이 발전함에 따라 탁상에 고정된 컴퓨터에서 계산되던 응용소프트웨어가 이동단말기로 옮겨가고 있다. 이로 인하여 등장하게 된 새로운 유형의 응용소프트웨어에는 위치기반서비스, 관광서비스, 이동전자상거래와 디지털 전투 등이 있고, 기존의 교통, 항공제어, 기상 예보, 119 서비스, 자원관리 서비스 등도 새로운 모습으로 거듭나고 있다. 이동물체 데이터베이스는 위에 열거된 서비스들을 가능하게 하여주는 열쇠가 되는 기술이다.

Wolfson은 이동물체 데이터베이스 분야에서 연구해야할 과제가 위치정보의 모델을 구축하고, 정확성을 관리하는 기술을

개발하고, 시공간 데이터를 접근하기 위한 언어를 개발하고, 검색을 위한 색인을 작성하는 방법과 데이터가 대용량으로 증가하는 것을 다루는 기술, 데이터 마이닝 기법, 서버의 과부하를 경감시키는 방법, 개인보호와 보안 등이라고 규정하였다[10].

위치정보를 나타내는 가장 간단한 방법으로 위치와 시각으로 구성된 순서쌍인 (loc, time) 형식을 들 수 있다. 순서쌍 (loc, time)은 time 시각에 이동물체의 위치가 loc 이었다는 사실을 나타내며, 이동물체의 위치를 주기적으로 측정하여 작성한다. loc는 x-y좌표나 위도-경도 좌표일 수 있으며, 측위 방법에 따라 셀 번호가 될 수도 있다. 이 방법을 본 연구에 적용한다면 화면상의 좌표가 될 것이다.

이 방법은 다음과 같은 단점을 갖는다. 첫째, 보간법을 사용할 수 없다. 예를 들어, 오후 3시에 주어진 지점에서 가장 가까이 있었던 이동물체가 무엇이나는 질문에 대한 답을 구하면 오후 3시 정각에 자기의 위치를 갱신한 이동물체 중에서만 구하게 된다. 따라서 만약 오후 3시 정각에 자신의 위치를 갱신한 이동물체가 없었다면 답을 구할 수도 없고, 오후 3시에 바로 그 자리에 있었던 이동물체가 있었다라도 그 이동물체가 자신의 위치를 갱신하지 않았다면 다른 이동물체를 답으로 구하게 된다. 나아가서 가까운 장래의 상황에 대한 질문에 대한 답을 구하기는 아예 불가능하다.

두 번째 단점은 첫 번째 단점과 밀접한 관계가 있다. 즉, 보간법을 사용할 수 없으므로 이동물체가 자신의 위치를 갱신하는 횟수를 증가시켜야 이동물체의 위치를 정확히 파악할 수 있게 된다. 갱신 횟수가 증가하면 필연적으로 데이터의 양이 급격히 증가하게 되고 처리를 위한 CPU 시간을 비롯한 많은 자원을 사용하게 된다.

끝으로 이 방법으로 데이터를 저장하면 이러한 데이터를 이용하는 응용소프트웨어 개발이 매우 불편하게 된다. 우선, 데이터베이스에 저장된 데이터는 갱신하기 전에는 변하지 않는다. 즉, 현실 세계의 이동물체는 부단히 움직이는데 반하여 데이터베이스에 기록된 이 이동물체의 위치는 갱신하지 않으면 변하지 않으므로 데이터베이스의 데이터와 실제 세계의 값이 서로 불일치하게 된다. 그리고 응용 프로그램에서 원하는 데이터를 찾아 주는 질의문을 작성하기가 매우 어렵다. 예를 들어, 원하는 데이터가 4시부터 5시 사이에 지정된 영역에 계속 머물러 있었던 이동물체일 때, 이러한 이동물체를 찾는 SQL 질의문을 작성하기가 매우 어렵다.

그럼에도 불구하고, (loc, time) 형식은 이동물체의 경로가 이미 알려져 있는 경우에 유용하다. 즉, 대중 교통 수단인 버스는 정해진 경로로 움직이기 때문에 버스의 시간-공간 정보는 (loc, time) 형식으로 표현해도 좋다. 또한, 이동물체의 지나온 궤적으로부터 미래의 위치를 예상하기가 불가능한 경우에도 이러한 형식을 사용할 수밖에 없다.

위치정보를 나타내는 다른 방법으로 궤적 모델을 은 제안한다. [10] 이 방법의 기본 전략은 이동물체의 목적지에 대한 이미 알려진 정보를 이용하는 것이다. 예를 들면, 아침 8시부터 9시 사이에 이 사람은 집에서 직장으로 출근한다든지 혹은 오

늘 스케줄에 의하면 이 사람은 오후 3시에 모차에서 이사회에 참석해야 한다든지 등의 부가적인 정보를 참조하는 것이다.

부가적인 정보에 의하여 이동물체의 목적지를 알고 있다면, 전자지도를 이용하여 이동물체의 궤적을 작성할 수가 있다. 전자지도는 관계형 데이터베이스의 관계(relation: 컴퓨터 과학 분야의 전공 용어임)로서 교차로에서 다른 교차로까지의 도로 세그먼트를 나타내는 터플(튜플이라고 발음하기도 함. 관계를 구성하는 한 줄을 일컬음)로 구성된다. 도로 세그먼트를 나타내는 전자지도의 속성에는 해당 도로 세그먼트를 지칭하는 유일한 번호와 도면상의 도로의 기하학적인 모양을 나타내는 일련의 직선들이 있다. 일련의 직선들은 물론 직선들의 시작점과 끝점으로 표현된다.

나아가서 좌표를 행정상의 주소로 변환하는 서비스를 가능하게 위하여 도로 세그먼트의 거리 이름, 좌측 거리 이름, 우측 거리 이름, 도로 유형이 남북 도로인지 동서 도로인지, 속도제한, 일방통행인지, 우편번호 등의 속성과 통과하는데 걸리는 시간 계산을 돕기 위하여 도로 세그먼트의 길이와 평균 통과 시간 등의 속성도 갖는다. 출발지와 목적지가 주어지면 이동물체의 경로는 출발지에서 목적지로 가는 최단경로를 이루는 일련의 도로 세그먼트들로 구성되며, 이를 바탕으로 주어진 시각에 이동물체가 어디에 위치하는지 예측할 수 있다.

궤적 모델을 이용하여 이동물체의 시공간 정보를 저장하면 다음과 같은 질의에 대한 답을 수월하게 구할 수 있다. 우선 하나의 궤적과 관련된 질의로 예를 들면 "이동물체 x가 주소 yy에 가장 가까워지는 시각은 언제인가?"와 같은 질의가 있다. 그리고 두 궤적 간의 관계를 묻는 질의로 예를 들면, "몇 시부터 몇 시 사이에 거리가 얼마 이내이던 이동물체에는 어떤 것들이 있는가?" 등이 있다.

이동물체의 위치 정보는 이동물체 자체가 부단히 움직이고 있을 뿐 아니라 측정 방법에도 오차가 내포되어 있기 때문에 어쩔 수 없이 어느 정도 부정확하다. 나아가서 질의 자체가 막연한 경우가 많아서, 예를 들면 "주어진 지점과 가까이 있는 이동물체들에는 어떤 것들이 있는가?" (가깝다는 말의 뜻이 막연함), 이동물체 데이터베이스는 불확실성을 다룰 수 있어야 한다. 불확실성을 다룰 때에는 불확실한 정도를 지칭하는 기준이 있어야 한다. 불확실한 정도를 너무 많이 주면 처리 대상이 되는 정보가 너무 많아지는 경향이 있고 불확실한 정도를 너무 적게 주면 타당한 답도 제외시키게 되는 결과를 낳게 된다. 그래서 적당한 불확실 정도를 계산하는 방법과 이 방법을 데이터 warehousing에 사용한 예가 예 소개된다.[11] 또한 "확률적으로 제한적인 직사각형"이라는 개념을 도입하여 조건에 맞는 데이터를 선택하고 그렇지 않은 데이터를 제거하는 방법이 예 소개된다.[12]

2. 칼만 필터

Extended Kalman Filter[13]는 상태를 나타내는 값을 구하는 과정으로 예측과 실측치를 이용한 갱신을 반복한다. 예를 들어 옥내 측위의 경우에는 단말기의 위치를 임의의 값으로

예측하고 실측값을 반영하여 갱신한 다음, 이를 토대로 또 다시 현재 위치를 예측하고 실측값을 반영하여 갱신하는 과정을 반복함으로써 단말기의 위치를 결정한다.

본 논문은 이동객체 데이터베이스의 위치 정보를 갱신하는 전략을 제안함으로 옥내 측위를 위한 칼만 필터를 소개하면 다음과 같다. 다양한 옥내 측위 방법이 있으나, 본 논문에서는 AP(Access Point)의 신호 세기를 바탕으로 위치를 판정한다고 가정한다. 모르는 위치의 이동 단말기 $A = [x \ y \ z]^T$ 에서 알고 있는 위치의 $AP_i = (X^i, Y^i, Z^i), i=1, \dots, m$ 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 거리 측정치를 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, $(h_x^i = \frac{X^i - x_0}{r_0^i}, h_y^i = \frac{Y^i - y_0}{r_0^i}, h_z^i = \frac{Z^i - z_0}{r_0^i})$ 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. 서있는 단말기의 위치를 판정하는 경우에는 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, $Q_k = 0$ 으로 멈추어 있음이 틀림없다고 모델링한다. 시스템 모델과 측정 모델을 일반화 하면 $X_{k+1} = X_k + W_k$, $Z_k = h(X_k) + V_k$ 이고, 칼만 필터를 적용하기 위하여 비선형 측정식을 선형화하면 $\delta X_{k+1} = \phi_k \delta X_k + W_k$, $\delta Z_k = H_k \delta X_k + V_k$ 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.

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

〈표 1〉은 EKF의 시스템 모델과 측정 모델을 정리하여 최종 위치를 구하는 과정이며, 정지 모델을 이용하므로 $\phi_k = I$, $Q_k = 0$ 으로 적용하면 〈표 1〉의 4), 5), 6)은 다음과 같이 정리된다.

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

$$P_{k+1} = (I - K_{k+1}H_k)P_k(I - K_{k+1}H_k)^T + K_{k+1}R_{k+1}K_{k+1}^T$$

표 1. 확장 칼만 필터 처리 과정
Table 1. The process of Extended Kalman Filter

1)Initial guess: \hat{X}_0 and P_0
2)Linearizing: $h(X_k) = h(\hat{X}_k) + H_k \delta X_k$, $\hat{Z}_k = h(\hat{X}_k)$
3)Kalman Gain: $K_{k+1} = P_k H_k^T (H_k P_k H_k^T + R_{k+1})^{-1}$
4)Measurement update: $\hat{X}_k = \hat{X}_k + K_{k+1}(Z_{k+1} - \hat{Z}_{k+1})$
5)Update error covariance: $P_k = (I - K_k H_k)P_k$
6)State propagation: $\hat{X}_{k+1} = \Phi \hat{X}_k$, $P_{k+1} = \Phi_k P_k \Phi_k^T + Q_k$
7)Goto step2

〈표 1〉에 나타나는 칼만 필터 과정은 오차공분산 초기값 (P_0), 프로세스 잡음 강도(Q), 측정 잡음 공분산값(R) 등의 파라미터에 적절한 초기 값을 부여함으로써 최적의 성능을 낼 수 있다. 따라서 이들 초기 값을 찾는 것이 중요하다.

III. 갱신 정책 제안

본 논문이 제안하는 갱신 정책은 칼만 필터를 이용한 무정보 추산 정책(KFDR: Kalman Filter Based Dead-Reckoning)

이다. 즉, 시점 t_0, t_1, \dots, t_i 각각에서 실측한 위치 정보에 칼만 필터를 적용하여 t_{i+1} 이후의 단말기의 위치를 추정하고, 추정된 위치와 실측 위치간의 차이가 일정 수준 이하이면 통신을 생략하는 전략을 쓰는 것이다.

이를 위하여 〈표 1〉에 보이는 측위 알고리즘을 우선 제적을 구하도록 변경한다. 이를 위하여 〈표 1〉은 위치를 3차원 공간에 나타냈지만, 본 논문에서는 2차원 공간에 나타낸다고 한다. 육내 위치기반 서비스에서는 서비스 공간이 평면이므로 2차원 공간이라고 가정하여도 무방하다. 나아가서 칼만 필터의 실측 단계에서 〈표 1〉은 AP까지의 거리를 실측하는데 반하여 본 논문에서는 K-NN 방법을 사용하여 실측 값 자체가 좌표가 된다고 한다. 따라서 〈표 1〉의 X, Φ, H 를 각각 다음과 같이 정한다.

$$X_k = \begin{bmatrix} x_k \\ y_k \\ v_{xk} \\ v_{yk} \end{bmatrix}, \Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

X_k 의 원소 x_k, y_k, v_{xk}, v_{yk} 는 각각 x좌표, y좌표, 이동객체의 x축 방향 이동 속도, y축 방향의 이동 속도를 나타낸다. Φ 의 Δt 는 k번째 위치 측정 후 경과 시간을 나타낸다. 예를 들어 매 1초마다 위치를 측정한다면 Δt 를 1로 교체한다. 〈표 1〉의 단계 6에서 $\hat{X}_{k+1} = \Phi \hat{X}_k$ 은 k번째 단계에서 구한 이동객체의 위치에 속도와 시간의 곱을 더하여 k+1 번째 위치를 예측한다. 본 논문이 제안하는 방법은 Δt 에 k번째 위치측정 후 지금까지 경과한 시간을 대입함으로써 현재의 위치를 추정하는 것이다.

일련의 위치 측정값들에 〈표 1〉에 보이는 칼만 필터를 적용하면 v_{xk} 와 v_{yk} 가 각각 일정한 값으로 수렴하게 된다. 따라서 본 논문이 제안하는 방법은 이동 객체의 위치를 비교적 정확하게 추정할 수 있다. 속도가 일정한 값에 수렴하였다는 사실은 〈표 1〉의 오차공분산 행렬인 P행렬의 원소를 보고 알 수 있다. 〈표 1〉은 초기에 X값을 임의로 결정하기 때문에 P행렬의 원소들의 값을 매우 큰 값으로 초기화한다. 칼만 필터의 반복 횟수가 증가하여, 이동 객체의 상태가 안정화되어 감에 따라 P의 원소들의 값은 점차 감소한다. 본 논문은 P의 원소들의 값이 일정 수준(예를 들면 0.001) 이하이면 이동 객체의 위치를 추정하기 시작할 것을 제안한다.

IV. 제안 알고리즘

육내 이동객체 데이터베이스 시스템의 핵심 구성 요소는 데이터베이스 서버, 웹 서버 그리고 통신 서버이다. 통신 서버는 이동 객체가 소지한 단말기가 송신한 단말기의 위치 정보를 수신하여 데이터베이스 서버에 전달하고, 웹 서버는 사용자가 입력한 명령을 데이터베이스 서버에 전달하며, 데이터베이스 서버가 작성한 답을 사용자의 단말기에 출력하여준다. 데이터베이스 서버는 단말기가 전송한 이동객체의 위치 정보와 적용 영역의 전자지도 정보를 저장하며 웹 서버가 전달한 질의어에 대

한 답을 구하는 역할을 담당한다. 이러한 옥내 이동 객체 데이터베이스의 구성은 <그림 1>에 보인다.

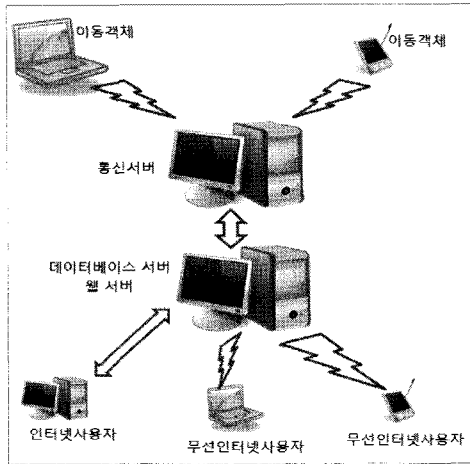


그림 1. 이동객체 데이터베이스 시스템 구성도
Fig. 1. Moving object database system configuration

본 논문은 이동객체와 통신 서버의 통신비용을 절감하기 위한 전략을 제안한다. 제안하는 전략은 시점 t_0, t_1, \dots, t_i 각각에서 실측한 위치 정보에 칼만 필터를 적용하여 t_{i+1} 이후의 단말기의 위치를 추정하고, 추정된 위치와 실측 위치간의 차이가 일정 수준 이하이면 통신을 생략하는 것이다. 단말기의 위치를 추정하는 시점은 P의 원소들의 값이 0.001 이하인 시점으로 한다. 이와 같은 전략을 수행하는 알고리즘은 <표 2>와 같다.

표 2. 제안하는 알고리즘
Table 2. Our algorithm for updating

- 1) 칼만 필터 알고리즘을 i 개의 측정치에 적용하여 \hat{X}_i 를 계산함
- 2) measurement = $i+1$ 번째 측정치 ;
- 3) if (elements of $P_0 < 0.001$) then {
 prediction = $\Phi \hat{X}_i$;
 if (|prediction - measurement| < 3)
 then skip sending
 else send measurement to the server
 else send measurement to the server; }

제안하는 알고리즘은 i 초 동안 매초마다 측정된 일련의 i 개의 측정치에 <표 3>에 보이는 칼만 필터를 적용하여, i 번째 교정치를 구한다. 그리고 P행렬의 원소들의 값이 모두 0.001 이하이면 Φ 의 Δ_i 를 1로 교체한 Φ 에 i 번째 교정치를 곱하여 현재 위치를 추정한다. 알고리즘에서는 추정치를 prediction이라는 변수에 저장하였다. 추정치와 측정치의 차이가 3미터 이하이면 측정치를 전송하지 않고, 그렇지 않은 경우에만 측정치를 전송하고 i 를 증가시킨다. 측정치를 전송하지 않은 경우에는 또 다시 1초가 경과한 후에 Φ 의 Δ_i 를 2로 교체한 Φ 에 i

번째 교정치를 곱하여 현재 위치를 추정한다.

<표 3>에 보이는 알고리즘은 <표 1>에 보이는 칼만 필터를 i 개의 측정치에 적용하여 궤적을 구한다.

표 3. i 개의 측정치로 궤적을 생성하는 칼만 필터
Table 3. Our Kalman filter procedure to produce a track with i measurements

```

algorithm 칼만궤적( $M_0$ )
//  $M_0$ ;  $t_0$ 에 획득한 측정치
1.초기화:  $X_{current} = M_0$  ( $M_0$ 는  $t_0$ 의 측정치),
    그리고  $P_0 = \begin{bmatrix} 300 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 \\ 0 & 0 & 300 & 0 \\ 0 & 0 & 0 & 300 \end{bmatrix}$ 
( for j=1; j <= i; j++ ) {
2..P_matrix = Phi_matrix * P_matrix *
    Phi_matrix.Transpose() + Q_matrix;
3.Matrix temp1 = new Matrix(2, 2);
4.Matrix temp2 = new Matrix(2, 2);
5.temp1 = H_matrix * P_matrix *
    H_matrix.Transpose() + R_matrix;
6.temp2 = temp1.Inverse();
7.K_matrix = P_matrix * H_matrix.Transpose()
    * temp2;
8.Z_matrix = new Matrix(2, 1);
9.Z_matrix = j 번째 측정치;
10.X_next = X_current + K_matrix *
    (Z_matrix - H_matrix * X_current);
11.X_current = Phi_matrix * X_next;
12.P_matrix = (L_matrix - K_matrix *
    H_matrix) * P_matrix;
}
end 칼만궤적
    
```

<표 3>의 $X_{current}$ 와 Φ 는 각각 <표 1>의 \hat{X} 과 Φ 에 해당한다. Q matrix(Q)와 R matrix(Q)의 초기 값은 다음과 같이 설정한다.

$$Q = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

데이터베이스 서버는 이동객체의 xx 시점의 위치를 묻는 질의가 들어올 때, xx 시점의 이동물체의 위치가 데이터베이스에 기록되어 있으면 이곳 위치를 질의의 답으로 반환한다. 그러나 xx 시점의 위치가 기록되어 있지 않으면 데이터베이스에 기록되어있는 xx 이전의 위치 정보들 중 가장 근접한 일련의 위치 정보를 추출하여 여기에 칼만 필터를 적용함으로써 xx 시점의 위치를 추정하고 이 추정치를 반환한다. 이동 객체도 <표 2>의 알고리즘을 적용하여 이와 동일한 방법으로 자기 자신의 위치를 추정할 값이 측정치와 3미터 오차 미만일 때 전

송을 하지 않았으므로, 데이터베이스 시스템이 반환한 값은 실 측정치와 3미터 오차 범위 안에 있게 된다.

V. 실험 결과

제안하는 방법의 성능을 정확히 분석하기 위하여 실제 시스템에서 제안하는 알고리즘을 적용하여 실시간으로 실험치를 구해야하지만, 본 논문에서는 <그림 2>에 보이는 실험 공간에 실선으로 보이는 동선을 따라 0.5미터/초 속도로 걸어가면서 매초 1회 K-NN으로 측정한 지점의 좌표를 실험 데이터로 사용하였다. 즉, 실시간으로 적용하지 않고, 미리 측정한 일련의 측정치에 제안하는 전략을 적용하여 통신량이 얼마나 줄는지 알아보았다.

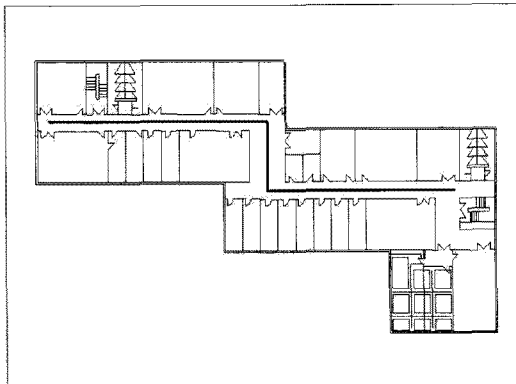


그림 2. 실험 공간과 실제 이동 경로
Fig. 2. Our test bed. The real track is shown as a solid line

이를 위하여 <그림 2>의 경로를 따라서 0.5미터/초 속도로 걸어가면서 매초 측위를 실시한 결과 총 142 개의 측정치를 얻게 됨을 알 수 있었다. 이와 같이 <그림 2>의 경로를 걸어가면서 매초마다 위치를 측정하는 작업을 25회 실시하였다. 각 회의 142번의 측정치에 제안하는 전략을 적용하여 전송을 스킵하는 횟수를 카운트한 후, 25회에 대한 평균을 구한 결과 약 74.6번(52.5%)을 전송하지 않고 예측 값으로 대체 할 수 있었다.

제안하는 전략이 오차에 미치는 영향을 알아보기 위하여 각 회의 142번의 측정치에 갈만 필터를 적용하여 구한 궤적과 실제 궤적 사이의 차이를 구한 결과 오차 평균이 2.924 미터이고, 클라이언트에 저장된 측정치와 추정치로 작성한 궤적의 오차는 평균 3.052 미터이며, 서버에 저장된 측정치와 추정치로 작성한 궤적의 오차는 평균 3.488 미터였다. 실험한 결과는 <표 4>에 정리하였다.

표 4. 실험 결과 요약
Table 4. A summary of our test results

	스킵 수	클라이언트 오차(mm)	서버 오차(mm)
실험1	73	2997.04	3199.14
실험2	78	2878.64	3195.85
실험3	77	3119.42	3493.37
실험4	74	3137.73	3709.39
실험5	75	3118.76	3864.57
실험6	77	3085.85	3642.85
실험7	71	3032.37	3173.09
실험8	71	3327.12	4316.38
실험9	69	3190.85	3231.05
실험10	78	3017.06	3311.39
실험11	73	3410.38	3636.03
실험12	79	3017.00	3800.37
실험13	77	3158.51	3409.40
실험14	78	2887.55	3487.09
실험15	73	3014.85	3398.04
실험16	77	3056.96	3686.33
실험17	76	3041.50	3730.61
실험18	74	2977.30	3678.67
실험19	73	3244.66	3874.83
실험20	71	2962.64	3121.75
실험21	74	3082.95	3387.70
실험22	76	2780.01	3263.12
실험23	76	2969.83	3226.30
실험24	72	2926.96	3288.63
실험25	73	2867.55	3095.06
전체 평균	74.6	3052.14	3488.84

참고문헌

VI. 결 론

본 논문은 옥내 이동객체 데이터베이스를 위한 갱신 전략을 제안하였다. 일련의 실측치에 제안한 전략을 적용한 결과와 적용하지 않은 결과를 비교한 결과, 제안하는 방법이 전송률을 47.5%로 줄여줌을 보였다. 전송율을 감축하여주는 대신 이동 단말기가 계산하는 이동 궤적의 오차는 약 4.4%가 증가하고 서버에서 계산하는 이동 궤적의 오차는 약 19.3% 증가함을 보였다. 추후에 제안하는 방안의 성능을 실제 상황에서 검증하는 실험을 수행할 예정이다.

- [1] Harter, A. and A. Hopper, "A New Location Technique for the Active Office", IEEE Personal Communications, Vol. 4, No. 5(1997), pp. 43-47.
- [2] Priyanthat, N., A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System", Proc. of 6th ACM International Conference on Mobile Computing and Networking, Boston, MA, Aug. 2000.
- [3] Want, R., A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System", ACM Transactions on Information Systems, Vol. 10, No. 1(1992), pp. 91-102.
- [4] Bahl, P. and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System", INFOCOM 2000, Mar. 2000, pp. 775-784.
- [5] Ito, S. and N. Kawaguchi, "Bayesian Based Location Estimation System Using Wireless LAN", Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, 2005. (PerCom 2005 Workshops) 8-12 March 2005, pp. 273 - 278.
- [6] Lassabe, F., P. Canalda, P. Chatonnay, and F. Spies, "A Friis-Based Calibrated Model for WiFi Terminals Positioning", Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2005, 13-16 June 2005, pp. 382-387.
- [7] Lin, T.N. and P.C. Lin, "Performance Comparison of Indoor Positioning Techniques Based on Location Fingerprinting in Wireless Networks", Proceedings of the 2005 International Conference on Wireless Networks, Communications and Mobile Computing, Vol. 2, 13-16 June 2005, pp. 1569-1574.
- [8] Yim, J. Park, C. Joo, J. Jung, S. "Extended Kalman Filter for Wireless LAN Based Indoor Positioning", Decision Support Systems 45 (2008), pp. 960-971.
- [9] Yim, J., Introducing a decision tree-based indoor positioning technique, Expert Systems with Applications 34 (2) 2008 1296-1302.
- [10] Wolfson, O., "Moving Objects Information Management: The Database Challenge", Proceedings of the 5th Workshop on Next Generation Information

- Technologies and Systems (NGITS'2002), Dan Caesarea Hotel, Caesarea, Israel, June 25-26, 2002, pp. 1-13.
- [11] Olston, C., Widom, J., "Offering a Precision-Performance Tradeoff for Aggregation Queries over Replicated Data", 26th International Conference on Very Large Data Bases, Cairo, Egypt, September 2000.
- [12] Tao, Y., Xiao, X., Cheng, R., "Range Search on Multidimensional Uncertainty Data", ACM Transactions on Database Systems, Aug. 2007, vol. 32, Issue 3, pp. 1-54.
- [13] Welch, G. and Bishop, G. "An Introduction to the Kalman Filter", Updated: July 24, 2006, <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>