

# 유비쿼터스 컴퓨팅 환경을 위한 실내 위치 추적 시스템의 설계

## A Design of Indoor Location Tracking System for Ubiquitous Computing Environment

우성현<sup>\*</sup>      전현식<sup>\*\*</sup>      김기환<sup>\*\*\*</sup>      박현주<sup>\*\*\*\*</sup>  
Sung-Hyun Woo    Hyeon-Sig Jeon    ki-Hwan Kim    Hyun-Ju Park

### 요 약

본 논문은 실내 환경에서 이동객체의 실시간 추적 알고리즘을 제안한다. 제안하는 시스템은 기존에 사용되던 삼각측량 기법과 DCM(Database Correlation Method) 기법을 통해 각각의 위치 데이터를 생성한 후, 그 중 이동객체와 더 근사한 위치 데이터를 실시간으로 선택한다. 또한 Kalman Filter를 사용하여 선택된 위치 데이터를 보정하여 시스템에 적용하므로 이동 객체의 위치 정확도를 향상시켰다. 기존에 연구된 Kalman Filter는 과거의 정보를 이용하여 현재의 위치를 추정해 내는 시스템의 특성상 안정화 되는 시간까지 불확실한 위치 데이터를 가지게 된다. 하지만 제안하는 위치 추적 시스템은 기존의 Kalman Filter를 그대로 적용하지 않고, 더 효율적인 방안을 제시한 후 적용함으로써 더 정확한 위치 추적을 가능케 한다.

### Abstract

This paper propose a realtime tracking algorithm of mobile object in indoor environment. this proposed system selects location data closer to mobile objects in real time that are results of Triangulation method and DCM(Database Correlation Method) method. Also, this system applies adjusted location data selected by using Kalman filter, and in result it improved location accuracy of transfer object. Be studied in existing the Kalman filter have unstable location data until its settlement because of it extracts current values by using the past the information. However, proposed location tracking system don't apply existent Kalman filter to this system and it permits precisional tracking location by uses more effective methods.

☞ Keyword : Location(위치), Tracking(추적), Kalman Filter(칼만필터), Triangulation(삼각측량), Fingerprint(핑거프린트)

## 1. 서 론

### 1.1 연구 배경

컴퓨터화의 새로운 컴퓨팅 패러다임으로 등장한

유비쿼터스화는 유비쿼터스 컴퓨팅(Ubiquitous Computing)과 유비쿼터스 네트워크(Ubiquitous Network)를 기반으로 물리공간을 지능화함과 동시에 물리공간에 펼쳐진 각종 사물들을 네트워크로 연결시키려는 노력으로 정의 할 수 있다[1]. 유비쿼터스 컴퓨팅과 유비쿼터스 네트워크란 물리공간에 보이지 않는 컴퓨터를 집어넣어 모든 사물이 네트워크로 연결되어 정보를 주고받는 것이다. 이는 모든 컴퓨터가 서로 연결되고 사용자 눈에 보이지 않으면서도 언제 어디서나 사용이 가능하고 현실세계의 사물과 환경 속으로 스며들어 일상생활에 통합되는 것을 전제로 한다. 현재 주목 받고 있는 홈네트워킹(Home Networking)이나 모바일 컴퓨팅

\* 준 회 원 : 한밭대학교 정보통신전문대학원 정보통신공학과, 석사 과정  
wsungh123@gmail.com(제1저자)

\*\* 정 회 원 : 한밭대학교 정보통신전문대학원 전파공학과, 박사 과정  
hsjeon@hanbat.ac.kr(제2저자)

\*\*\* 정 회 원 : 삼성전기 무선통신 솔루션 사업부 전임 연구원  
kihwan1224.kim@samsung.com(제3저자)

\*\*\*\* 정 회 원 : 한밭대학교 정보통신컴퓨터공학부, 부교수  
phj@hanbat.ac.kr(교신저자)

[2006/03/15 투고 - 2006/03/21 심사 - 2006/05/22 심사완료]

(Mobile Computing) 같은 기술보다 한 단계 발전된 컴퓨팅 환경을 말하며, 특히 이를 매개로 언제 어디서나 사람 혹은 사물의 위치정보를 얻을 수 있고 그에 적합한 정보를 제공할 수 있다. 실내 환경에서의 위치인식 기술은 유비쿼터스 시대에 주목받고 있는 가장 핵심이 되는 분야로 실내 환경에 적합한 위치 추적 시스템에 관한 연구가 최근 부각되고 있다. 그 중 삼각 측량 기법을 사용한 위치 추적 시스템이 대표적이라 할 수 있다. 하지만 삼각 측량 기법은 TOA(Time of Arrival)를 사용하기 때문에 건물의 외벽이나 창문의 메탈 성분 같은 장애물로 인한 전파의 반사와 회절, 분산 등의 NLOS (Non-Line-Of-Sight) 환경으로 인하여 정확한 위치추적이 어렵다는 문제점이 있다[2]. 이러한 NLOS 문제를 완화한 정확한 위치추적을 위해 기존 연구에서는 삼각 측량 기법을 통해 생성된 위치 데이터에 데이터베이스를 활용한 위치 추정 시스템인 DCM(Database Correlation Method)과 위치 보정 알고리즘인 Kalman Filter와 같은 기법들의 이용을 제안하고 있다. 하지만 현재 사용되고 있는 실내 환경에서의 DCM 기법은 환경의 변화에 능동적으로 대처하기 어렵다는 문제점이 존재한다. 따라서 DCM을 사용한 위치 추적 기법은 환경의 변화에 따라 부정확한 위치 데이터를 생성하게 되고, 이로 인하여 삼각 측량 기법을 사용한 위치 추적 시스템보다 부정확한 오차를 갖는 경우가 빈번하게 생긴다는 문제점이 있다. 또한 기존에 연구된 Kalman Filter는 과거의 정보를 이용하여 현재의 값을 추정해 내는 시스템의 특성상 안정화 되는 시간까지 불확실한 위치 데이터를 가진다는 문제점이 존재한다.

## 1.2 연구 목적 및 논문의 구성

앞서 언급한 것과 같이 기존의 실내 위치 추적 기법들은 성능향상을 위해 개선되어야 할 문제점들을 가지고 있으며, 이 문제점들을 효과적으로 개선한다면 효율적이고 실용 가능한 실내 위치 추적 방법

이 될 수 있다. 따라서 본 논문에서는 효율적이고 실용 가능한 실내 위치 추적을 위하여 기존에 사용되던 삼각측량 기법과 DCM 기법을 통해 각각의 위치 데이터를 생성한 후, 그 중 이동객체의 실제 위치와 더 근사한 위치 데이터를 실시간으로 선택하여 시스템에 적용하는 방안을 제안한다. 또한 기존 Kalman Filter의 문제점을 해결하는 방법을 제시한 후, 선택된 위치 데이터에 제안하는 Kalman Filter 기법을 적용하여 보정함으로써 이동 객체의 위치 정확도를 향상시키는 방안에 대하여 연구한다.

본 논문의 구성은 총 5 장으로 구성되어 있으며, 그 내용은 다음과 같다. 2 장에서는 본 논문과 밀접한 관련이 있는 삼각 측량 기법과 DCM 알고리즘에서 가장 대표적인 기법인 Fingerprint 기법, 그리고 Kalman Filter에 대해 기술한다. 3 장에서는 제안하는 실내 위치 추적 시스템(ILTS : Indoor Location Tracking System)을 소개하고, 시스템의 구조를 설명한다. 그리고 위치 추적 알고리즘을 기술한다. 4 장에서는 제안하는 위치 추적 시스템의 성능 분석을 통한 연구 결과를 기술하고, 5 장에서는 결론을 맺으며, 향후 연구 과제를 기술한다.

## 2. 관련 연구

### 2.1 삼각 측량 기법 (Triangulation technique)

본 절에서는 제안하는 실내 위치 추적 시스템에서 위치 추적을 위해 사용되는 삼각 측량 기법에 대해서 소개한다. 삼각 측량 기법이란, 신호원과 수신기 사이의 전파 도달 시간(TOA : Time of Arrival)을 측정하여 신호원과 수신기 간의 거리를 알게 되는 측위 방식이다[3]. 일반적으로 분산된 많은 이동 객체(receiver)들은 server에서 전송된 신호를 받게 사용되고, TOA나 TDOA(Time Difference of Arrival)의 정보를 활용하여 receiver의 정확한 위치추적을 한다. TOA 기반의 receiver 위치 계산 방법은 TOA 기반 알고리즘들에 대한 원들의 교차점

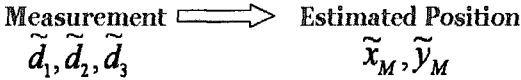
3 anchors with known positions (at least) are required  
Retrieve a 2D-position from 3 ToAs

$$d_1 = \sqrt{(x_1 - x_M)^2 + (y_1 - y_M)^2}$$

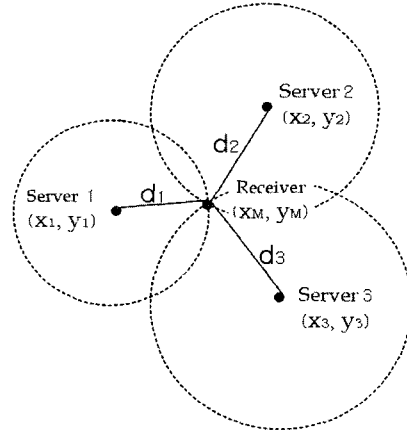
$$d_2 = \sqrt{(x_2 - x_M)^2 + (y_2 - y_M)^2}$$

$$d_3 = \sqrt{(x_3 - x_M)^2 + (y_3 - y_M)^2}$$

### Specific Positioning Algorithms



〈그림 1〉 TOA 신호를 사용한 Positioning



〈그림 2〉 삼각 측량 기법 도시화

을 계산하기 위해 기하 해석을 한다[4].

그림 1에서 보는 바와 같이 TOA는 receiver로부터 최소 3개의 server까지의 절대적인 시간을 계산하여 위치를 추적하는 알고리즘으로 동기식, 또는 비동기식으로 계산될 수 있다. 동기식 방법은 receiver와 server가 서로 동기화되어 있는 시스템이다. 이 방법에서 터미널은 server로 절대적인 현재 시간을 기록하여 신호를 보낸다. 이 신호는 receiver의 주변에 있는 각각의 server로 도달하게 된다. server와 receiver는 현재 시간을 동기적으로 맞추고 있기 때문에 server는 받은 신호에 대한 절대적인 도달 시간을 계산할 수 있으며, 이 시간에 대한 거리를 이용하여 receiver에 대한 위치를 계산하게 된다.

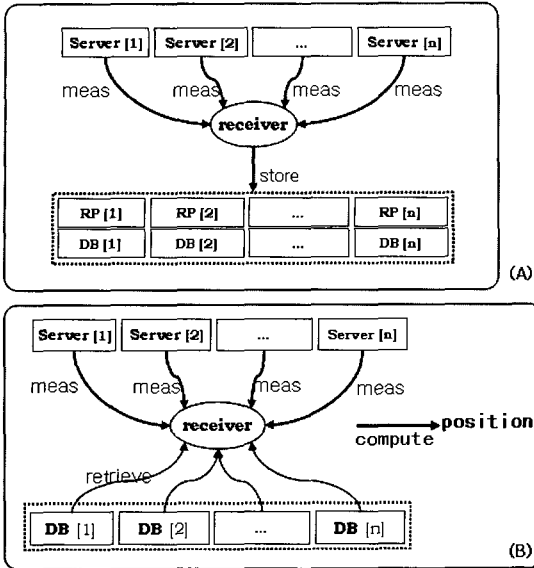
비동기식 방식에서는 모든 server와 receiver 간에 동기를 맞추는 필요가 없다. server는 현재 시간을 기록하고 receiver에 신호를 보낸다. receiver는 신호를 받는 즉시 응답 신호를 보내어 server에 도달하도록 한다. server는 receiver에서 보내온 신호의 도착 시간과 기존에 저장해 놓은 신호의 시간차를 이용하여 그림 1과 같이 거리를 계산하게 된다. 다른 server들도 같은 방법으로 거리를 계산한다. 그리고 구해진 거리 정보를 통하여 그림 2와 같이 receiver의 현재 위치 정보를 생성하게 된다[5].

## 2.2 Fingerprint

본 절에서는 DCM 알고리즘에서 가장 대표적인 기법인 Fingerprint를 소개한다. Fingerprint는 휴리스틱 정보를 데이터베이스에 저장한 후 receiver에서 입력된 위치 데이터 값과 비교하여 현재 위치를 return하는 기법이다[6].

이해를 돕기 위해 WLAN(Wireless Local Area Network) 환경에서 구축된 Fingerprint 시스템을 예로 들어 설명한다. WLAN 환경에서는 receiver의 위치를 측정하기 위해 각 AP(Access Point)로부터의 위치 정보(SS: Signal Strength)로 receiver의 위치를 측정한다.

그림 3에서 보는 것과 같이 Fingerprint는 Database(DB)를 생성하는 ‘Training’ 단계와 receiver가 요구하는 위치 데이터 값을 측정하는 ‘Positioning’ 단계로 나누어진다. ‘Training’ 단계는 측정된 위치를 참조할 Fingerprint 데이터베이스를 생성하기 위해 많은 요청이 발생할 것이라 판단되는 지점이나 사용자의 이동이 빈번한 위치를 RP(Reference Point)로 선택한다. 선택된 RP에 receiver를 위치시킨 후 모든 server들로부터의 SS를 측정한다. 그리고 각 server로부터의 위치 데이터를 Database에 기록한다. ‘Training’ 초기에 선택한 모든 RP를 방문할 때까지 반복 수행하여 유지한



〈그림 3〉 Training 단계(a)와 Positioning 단계(b).

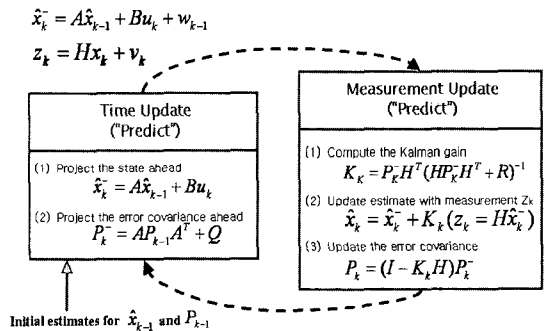
다[7]. ‘Positioning’ 단계는 receiver의 위치를 측정하기 위해 위치를 요구하는 장소에서 SS를 측정한다. 그 후 ‘Training’ 단계에서 유지된 Fingerprint Database와 비교한 후 요구한 receiver의 위치와 가장 적합한 위치 데이터의 정보를 receiver로 return 한다.

하지만 앞에서 살펴본 것처럼 Fingerprint는 측정 환경이 변화하게 되면 환경 변화 정보를 반영할 수 없기 때문에 실내 환경처럼 빈번한 환경 변화가 일어날 경우 환경 변화에 능동적으로 대처하기 어려운 문제점이 있다.

### 2.3 Kalman Filter

본 절에서는 제안하는 위치 추적 시스템에서 위치 보정을 위해 사용되는 Kalman Filter를 소개한다. Kalman Filter는 프로세스의 상태를 추적하기 위해 발생된 오차를 최소화 하는 방법으로 효율적인 계산 수단을 제공하는 수학 방정식의 집합이다 [8]. 이 필터는 신호와 잡음의 분리가 우수하여 순간적인 위치의 점프(JUMP) 현상을 방지할 수도 있으며, 정확한 위치, 속도 및 시간을 안정적으로

집출하는데 크게 기여한다. 또한 과거와 현재뿐만 아니라 미래 상태까지도 지원을 하며, 모델시스템의 특성이 자세히 알려지지 않은 경우에도 추적이 가능하다[9][10]. 이는 피드백 제어(feedback control) 형식을 사용하여 추적한다. 필터는 임의 시간에 프로세스 상태를 추적하고 노이즈 측정 형식으로 피드백을 얻는다. Kalman Filter의 평가 알고리즘은 수치 문제를 풀기 위한 예측자-교정자(predictor-corrector) 알고리즘으로, 그림 4에서 보는 바와 같이 시간에 앞서 미리 현재 상태를 예측하는 시간 갱신 방정식(Time Update Equation)과 해당 시간에 실제 측정에 의해 전달된 추정 상태 값을 조정함으로 피드백을 책임지는, 교정을 위한 측정 갱신 방정식(Measurement update equations)으로 구성된다.



〈그림 4〉 Kalman Filter의 operation Procedure

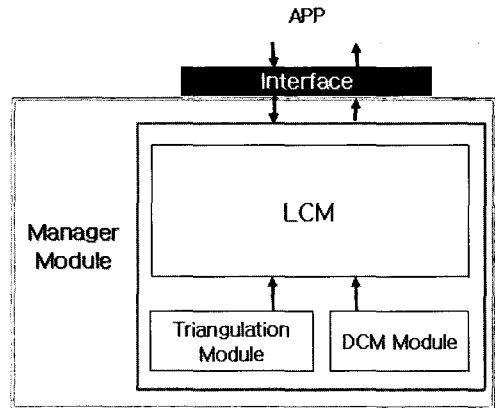
시간 갱신 방정식은 현재의 상태를 시간의 순방향으로 전달하고, 다음 시간 단계의 추적을 얻기 위한 에러 공분산을 추정한다. 그림 4의 “Time Update”의 단계에서 ‘x’는 최적화를 하고자 하는 상태 변수를 의미하며, 계수 ‘A’는 한 단계에서의 상태변수와 다음 단계의 상태변수를 연결하는 변환 계수를 표현한다. ‘B’와 ‘u’는 하나의 값으로 인식할 수 있으며 이들은 시스템에 무관한 추가 입력 값이다. 그리고 ‘w’는 k 단계에서 상태변수 ‘x’의 참값과의 차이 값, 즉 시스템 오차(system noise)이다. ‘w’는 Kalman

Filter 내에서 분산의 형태로 'Q'라는 변수로 적용된다. "Measurement Update"의 단계에서 'z'는 관측값이고, 이는 상태변수 'x'와 변환계수 'H'에 의해 표현되며, 'v'는 관측값 'z'와 관측 참값과의 오차(measurement noise)이다. 'v'는 'w'와 마찬가지로, 관측 참값에 대한 분산인 'R'이라는 변수로 Kalman Filter 내에 적용된다. 즉, 시간 갱신 방정식에서의 오차 'w'와 측정 갱신 방정식에서의 'v'는 각각의 참값에 대해 정규분포하며 그 평균은 0이고, 분산은 각각 'Q'와 'R'이다[8]. 그림 4에서 나와 있듯이 측정 갱신 방정식에서 첫 번째 단계는 Kalman gain 'K<sub>k</sub>'를 계산한다. 다음 'Z<sub>k</sub>'를 얻기 위해 실제 프로세스를 측정 후 측정을 통합하여 posteriori 상태 추정값을 얻는다. 이 작업은 새로운 값을 얻기 위해 반복된다. 마지막 과정은 posteriori 에러 공분산을 얻기 위한 단계이다. 각 시간과 측정갱신 쌍들 이후의 프로세스는 새로운 priori 추정을 투입 또는 예측을 이용한 이전단계 posteriori 추정들로 반복된다. 이 반복 특성은 Kalman Filter의 매력적인 특성 중의 하나로 각 추정에 대한 모든 데이터를 직접 연산하도록 설계된 Wiener 필터보다 구현을 훨씬 쉽게 한다. 하지만 과거의 정보를 활용하여 현재의 위치를 예측하는 시스템의 특성상, 초기 시스템의 경우 보정된 정보가 안정화 되는 시간까지 불확실한 위치 데이터를 감수해야 한다는 문제점이 존재한다[11].

### 3. 제안하는 실내 위치 추적 시스템 (ILTS : Indoor Location Track System)

#### 3.1 ILTS 시스템 구조

본 논문에서 제안하는 실내 위치 추적 시스템의 구조는 그림 5에서 보는 바와 같이 총 4개의 모듈로 구성된다.



(그림 5) ILTS의 구조

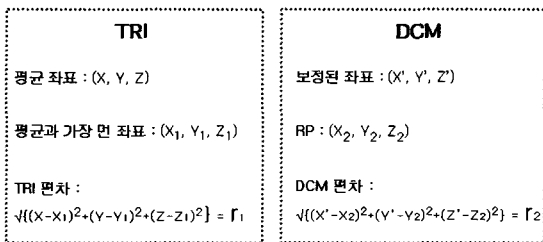
첫 번째 모듈은 전체 프로세스의 제어를 담당하는 MM(Manager Module)으로 위치 추적 프로세스를 총괄한다. 다음으로 위치 추적을 담당하는 TRI(Triangulation Module)와 DCM(DCM Module)으로 구성된다. 그리고 마지막으로 선택 알고리즘과 보정 알고리즘을 수행함으로써 최종 원하는 위치 데이터를 얻는 LCM(Location Correction Module)으로 이루어져 있다.

#### 3.2 ILTS 시스템의 흐름

제안하는 실내 위치 추적 시스템의 흐름은 다음과 같다. 먼저 TRI는 TOA로부터 생성된 거리 정보를 기반으로 기존의 삼각측량 알고리즘을 활용하여 receiver의 위치를 계산한다. DCM은 TRI 데이터를 활용하여 저장되어 있는 Database에서 가장 적합한 RP를 선별한 후 'Location Algorithm'을 통해 receiver의 위치를 계산한다. TRI와 DCM 연산 후 LCM에 각 Module에서 생성된 위치 데이터를 전송한다. LCM은 두 Module에서 입력 받은 데이터를 활용하여 실제 receiver의 위치 데이터와 더 근사한 위치 데이터를 선택한다. 그리고 선택된 위치 데이터를 Kalman Filter를 사용하여 보정한 후 MM에 데이터를 리턴한다. MM에서는 보정된 위치 데이터를 상위 APP(Application)로 전송한다.

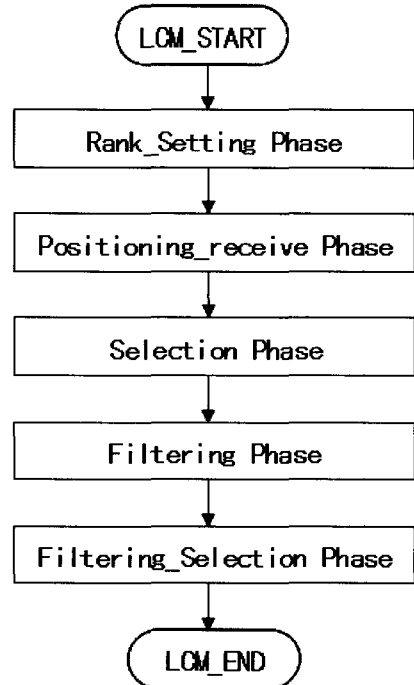
### 3.3 LCM (Location Correction Module)

본 절에서는 제안하는 ILTS의 핵심 모듈인 LCM을 소개한다. 앞에서 살펴 본 바와 같이 NLOS 환경과 실내에서의 빈번한 환경 변화 등으로 인하여 기존의 위치 추적 기법들은 정확한 위치 추적이 어렵다. 따라서 일반적인 위치 추적 기법을 사용한 TRI와 DCM의 위치데이터를 바로 적용하여 사용하기에는 무리가 있다. 따라서 LCM은 TRI와 DCM으로부터 수신된 위치 데이터를 비교 및 분석함으로써 실제 receiver의 위치와 더 근사한 위치 데이터를 선택하고, 선택된 위치 데이터를 Kalman Filter를 수행하여 보정함으로써 최종 receiver의 위치 데이터를 얻는 알고리즘을 수행한다.



〈그림 6〉 LCM에 수신되는 위치 데이터

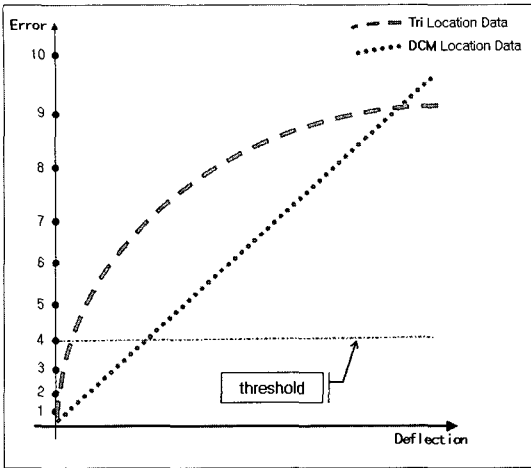
그림 6은 LCM에 수신되는 각 모듈의 위치 데이터이다. 그림 6에서 보는 바와 같이 TRI에서는 TOAs 정보를 활용하여 계산된 ‘평균 좌표’와 ‘평균과 가장 먼 좌표’, 그리고 두 정보를 사용하여 계산된 ‘TRI 편차’의 정보가 LCM에 수신된다. DCM에서는 TRI 위치 데이터와 RP를 활용하여 얻어진 ‘보정된 좌표’와 ‘RP 좌표’, 그리고 두 정보를 활용하여 계산된 ‘DCM 편차’의 정보가 LCM에 수신된다. LCM에서 ‘TRI 편차’ 정보를 수신하는 이유는 시스템의 특성상 ‘TRI 편차’는 실제 receiver의 좌표와 TRI에서 생성된 좌표와의 오차와 밀접한 관련성을 가지기 때문이다. 즉, TRI의 편차 데이터가 클수록, 오차도 커지게 된다. LCM에서 ‘DCM 편차’를 수신하는 이유 역시 ‘TRI 편차’를 수신하는 이유와 동일하다. 그림 6에 나타난 각 위치 데이터를



〈그림 7〉 LCM의 흐름도

수신하게 되면 LCM은 그림 7에서 보는 것처럼 다섯 단계의 과정을 수행하게 된다.

LCM은 가장 먼저 ‘Rank\_Setting Phase’를 수행하게 된다. 이 단계는 시스템 적용 전에 시행하는 일종의 Training 단계로, 실제 적용 환경에서 약 5,000~50,000개의 receiver의 real 위치 좌표와 TRI, DCM의 위치 데이터를 생성한 후 LCM에 수신한다. 그리고 수신된 TRI와 DCM의 ‘평균 좌표’와 ‘보정된 좌표’, ‘real 좌표’와 ‘TRI, DCM의 편차 데이터’를 활용하여 각 위치 데이터의 등급을 생성한다. 즉, receiver의 ‘real 좌표’와 ‘DCM의 보정된 좌표’와의 차인 ‘오차 데이터’와 DCM으로부터 수신된 ‘DCM 편차 데이터’를 통해 DCM 등급 분석을 위한 데이터를 생성한다. 그리고 receiver의 ‘real 좌표’와 ‘TRI의 평균 좌표’와의 ‘오차 데이터’와 TRI로부터 수신된 ‘TRI 편차 데이터’를 활용하여 TRI 등급 분석을 위한 데이터를 생성한다. 이를 통해 그림 8과 같이 그래프로 표현 가능한 규칙성이 존재함을 확인할 수 있다.



〈그림 8〉 LCM의 위치 데이터 선택을 위한 Rank Setting 그래프

이 규칙성을 바탕으로 표 1과 같이 등급 설정이 가능하다. 각 위치 데이터의 편차정보는 10 rank로 등급화 한다. 등급 설정 후 ‘threshold’를 설정한다. DCM의 위치 데이터는 TRI의 위치 데이터에 RP를 사용한 데이터이기 때문에 TRI에서 수신된 위치 데이터보다 receiver의 위치와 더 근사한 데이터일 가능성이 크다. 따라서 LCM의 세 번째 단계인 ‘Selection Phase’에서 TRI와 DCM의 위치 데이터 중에 수용 가능한 오차 범위를 ‘threshold’로 설정하여, ‘threshold’ 이하에서는 항상 DCM이 선택되도록 한다.

〈표 1〉 등급 설정 예 [단위 : mm]

RANK	오차	TRI 편차	DCM 편차
1	[0~100)	[0~1321)	[0~2862)
2	[100~300)	[1321~1380)	[2862~3298)
3	[300~600)	[1380~1520)	[3298~3794)
4	[600~1000)	[1520~1698)	[3794~4398)
5	[1000~1500)	[1698~1890)	[4398~5030)
6	[1500~2100)	[1890~2098)	[5030~5843)
7	[2100~2800)	[2098~2532)	[5843~6790)
8	[2800~3600)	[2532~3408)	[6790~7982)
9	[3600~4500)	[3408~4620)	[7982~9342)
10	[4500 ~ ∞)	[4620 ~ ∞)	[9342 ~ ∞)

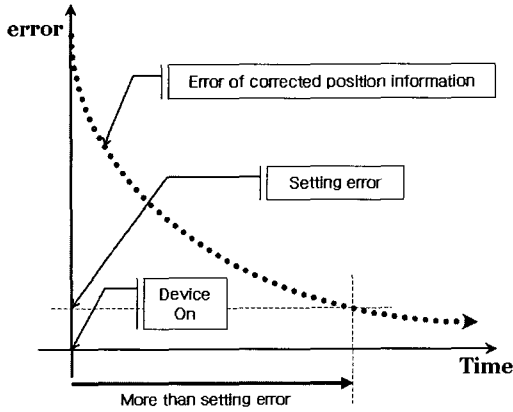
‘Rank\_setting Phase’ 후 실제 환경에서 적용될 ‘Positioning\_receive Phase’를 수행한다. ‘Positioning\_receive Phase’는 receiver가 이동함으로써 인하여 발생하는 TOA 정보를 활용하여 TRI의 위치 데이터를 생성하고, 생성된 TRI의 위치 데이터를 사용하여 DCM의 위치 데이터를 생성한 후, 생성된 각 위치 데이터를 LCM에 송신한다.

‘Positioning\_receive Phase’수행 후 ‘Selection Phase’를 수행한다. ‘Selection Phase’에서는 LCM에 송신된 TRI와 DCM의 위치 데이터 중 실제 이동객체인 receiver의 위치와 더 근사한 위치 데이터를 선택하는 알고리즘을 수행한다. LCM에 수신된 TRI와 DCM의 위치 데이터 중 편차 데이터를 표 1의 등급 설정 표에 대조한 후, TRI와 DCM의 해당하는 등급을 구할 수 있다. 앞에서 설명했듯이 DCM의 위치 데이터는 TRI의 위치 데이터에 RP를 사용한 데이터이기 때문에 TRI에서 수신된 위치 데이터 보다 더 정확한 데이터일 가능성이 크다. 따라서 편차 등급이 같거나, DCM의 편차 등급이 시스템에 맞게 미리 설정된 ‘threshold’ 이하, 즉 4 rank 이하일 시 항상 DCM 위치 데이터를 선택한다. 4 rank 이후부터는 TRI와 DCM의 편차 Rank를 비교하여, 등급이 서로 다른 경우 더 낮은 rank의 데이터를 선택함으로써 실제 receiver 위치와 더 근사한 위치 데이터를 선택한다.

다음으로 수행되는 ‘Filtering Phase’는 ‘Selection Phase’에서 선택된 위치 데이터를 Kalman Filter를 적용하여 보정된 위치 좌표를 생성한다. 하지만 앞에서 살펴본 바와 같이 Kalman Filter는 과거의 정보를 이용하여 현재의 값을 추정해 내는 특성상 예러 오차가 최소화된 보정 값을 얻기 위해서 Device가 가동 시 마다 안정화 되는 시간까지 불안정성을 가지게 된다. 예를 들어 Device를 최초 작동할 경우, 일정 시간 과거의 정보가 존재하지 않으므로 실제 위치 데이터와 수렴하는 일정시간 동안 큰 오차가 발생한다.

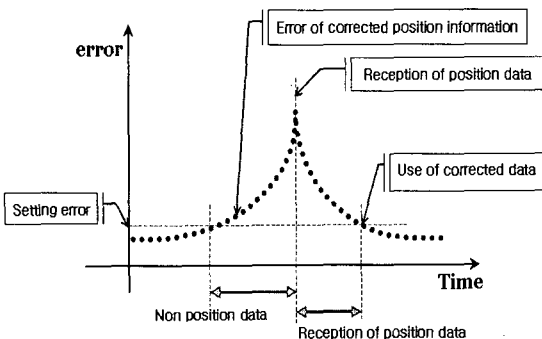
따라서 LCM에서는 이러한 단점을 해결하기 위하여 ‘Filtering Selection Phase’를 수행하게 된다.

‘Filtering Selection Phase’에서는 ‘Filtering Phase’에서 Kalman Filtering된 위치 데이터를 항상 최종 receiver의 위치로 사용하지 않고, 두 가지 Filtering 사용 여부 판단 알고리즘을 수행한 후 최종 receiver의 위치를 결정한다.



〈그림 9〉 Device를 가동시킨 상황에서 Filtering 사용 여부

첫째로, Device를 가동시킨 상황에서 Filtering 사용 여부를 판단한다. 그림 9에서 보는 것처럼 이동 객체가 Device를 작동했을 시 실제 사용할 위치 추적 시스템에 맞게 설정된 수용 가능한 ‘설정 오차’ 이상의 오차를 갖는 시간범위 내에서는 Kalman Filter의 보정된 위치데이터를 사용하지 않고, 대신 ‘Selection Phase’에서 선택된 위치 데이터를 MM에 전송한다.



〈그림 10〉 위치데이터 수신 시간간격에 따른 필터링 사용 여부

두 번째로, 위치 데이터 수신 간격에 따른 Filtering 사용 여부를 판단한다. 그림 10에서 보는 바와 같이 LCM에 일정시간 Device의 위치 데이터가 수신되지 않을 경우, 정확한 위치 데이터를 나타내는데 어려움이 있다. 하지만 Kalman Filter는 피드백 제어(feedback control) 형식을 사용하여 추적하므로, 과거와 현재뿐만 아니라 미래 상태까지도 지원을 하며, 모델시스템의 특성이 자세히 알려지지 않은 경우에도 추적이 가능하기 때문에, LCM에서는 일정 시간 Device의 위치정보가 수신되지 않을 경우에 기존의 과거정보를 이용하여 계속적으로 Kalman Filter를 이용한 보정 과정을 반복함으로써 보정된 위치 데이터를 선택하여 MM에 전송한다. 하지만 오랜 시간동안 위치 데이터가 수신되지 않을 경우에 의존할 수 있는 위치 데이터가 존재하지 않으므로, 그에 따라 발생하는 오차는 감수하여야 한다. 일정 시간 후 LCM에 위치 추적 시스템의 위치 데이터가 수신될 시, 기존 과거정보에 의존하여 보정된 위치 데이터는 삭제되고, 앞에서 설명한 첫 번째 Filtering 사용 여부 판단 알고리즘을 수행한다. 이를 통해 기존의 Kalman Filter가 가지는 문제점을 최소화하고, 정확한 이동객체의 위치 추적을 가능케 한다.

#### 4. 성능 평가

본 장에서는 3장에서 제안한 ILTS의 효율성을 평가하기 위해 성능 분석 결과를 제시한다. 그리고 기존 실내 위치 추적 시스템과의 비교한 후 구현한 시스템의 장점을 기술한다.

##### 4.1 실험 환경

모의실험 환경 조건은 표 2와 같다. receiver의 위치 측정은 1초에 3번 측정하는 것을 가정하고, random 데이터를 생성하여 receiver의 움직임을 시뮬레이션 하였다. 총 10만회 테스트 수행 후 Window Size를 50으로 설정하여 샘플링한 결과를 도시하였다.



<표 2> 모의실험 조건 (Parameter)

receiver의 수	1개
server의 수	5개
RP의 수	9개, 18개, 36개
receiver의 시작 위치	[X]15,000 [Y]15,000 [Z]1,000
면적	30 X 30 m <sup>2</sup>
객체 이동속도	150cm/sec and 300cm/sec
거리 데이터	3m의 오차를 가짐

모의실험을 수행한 컴퓨팅 환경은 표 3과 같다.

<표 3> 실험 컴퓨팅 환경

CPU	AMD 2500+
RAM	DDR3200 512MB
Operating System	Microsoft Windows XP sp2

## 4.2 실험 결과

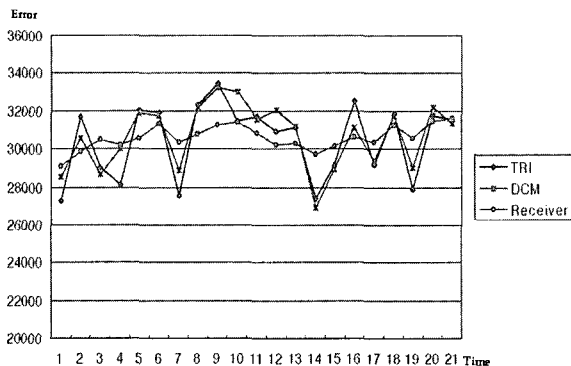
그림 11은 window Size를 20으로 설정하여 샘플링한 결과를 도시화하여 TRI 위치 데이터와 DCM 위치 데이터와 실제 receiver의 위치를 표현하였다.

그림 11에서 보면 알 수 있듯이, DCM은 TRI의 위치데이터와 데이터베이스에 저장된 RP

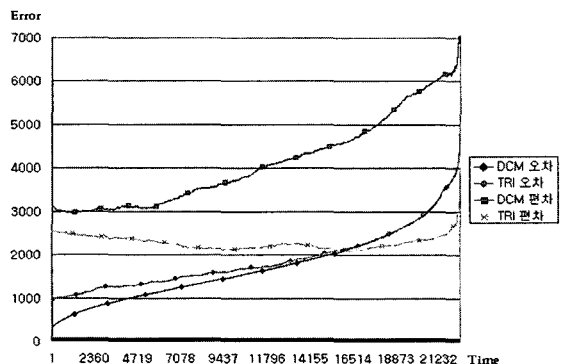
를 사용하여 위치 보정을 하였음에도 불구하고, 실내 환경처럼 빈번한 환경 변화가 발생할 경우 환경 변화에 능동적으로 대처하기 어려운 문제점으로 TRI보다 오차가 더 큰 경우가 간혹 발생한다. 따라서 ILTS는 DCM과 TRI의 위치데이터를 'Rank\_Setting Phase'에서 생성된 rank에 적용함으로, receiver에 더 근사한 위치데이터를 선택하게 된다.

그림 12는 'Rank\_Setting Phase'를 통해 생성된 분석 그래프로, 각 위치데이터의 편차와 오차간의 관계를 보여준다. 그림 12를 살펴보면, DCM의 편차가 클수록 DCM의 오차도 같은 형태로 커지는 것을 볼 수 있다. TRI는 작은 오차 범위 내에서 편차와 오차간의 상관성이 존재하지 않지만, 일정 오차 이상, 즉 TRI 오차가 200mm에서부터 두 관계간의 상관성이 이루어지는 것을 살펴볼 수 있다. 이 관계를 통해 그림 13과 같은 편차-오차 그래프의 생성이 가능하다.

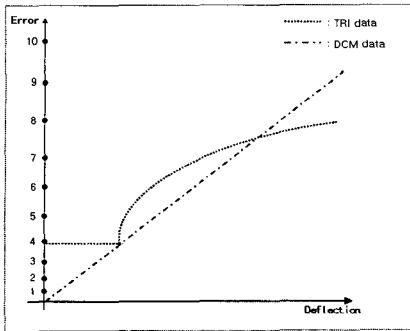
그림 13의 'Rank\_Setting Phase'에서 생성된 Rank Setting 그래프를 활용하여 'Selection Phase'에서 receiver에 더 근사한 위치데이터를 선택하고, 선택된 위치 데이터를 Kalman Filter를 적용하여 보정함으로 그림 14에서 보는 바와 같이 receiver와 더 근사한 위치 데이터를 얻었음을 확인 할 수 있다.



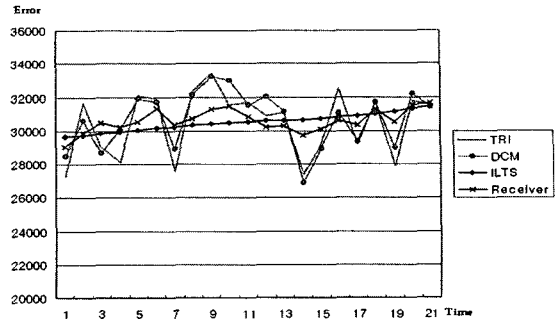
<그림 11> TRI, DCM 성능 비교



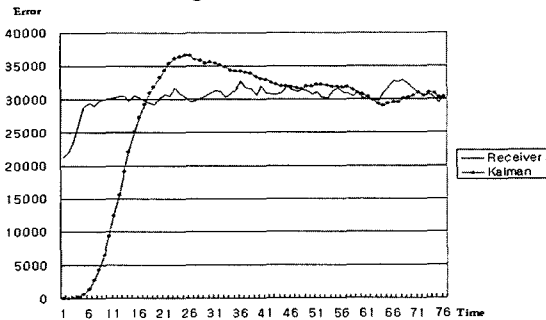
<그림 12> 각 위치데이터의 편차와 오차간의 관계 그래프



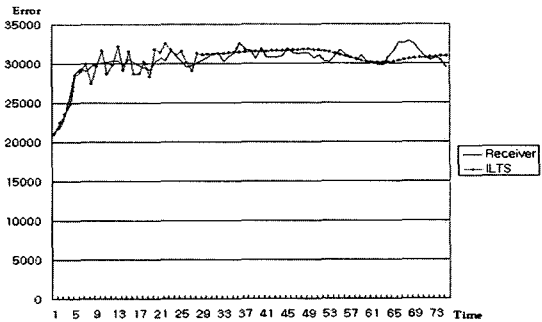
〈그림 13〉 LCM의 위치 데이터 선택을 위한 Rank Setting 그래프



〈그림 14〉 ILTS 수행 결과



〈그림 15〉 디바이스 최초 작동 시 Kalman Filter 수행 결과



〈그림 16〉 Device 최초 작동 시 ILTS 수행 결과

〈표 4〉 위치 추적 시스템의 10만회 수행 결과 (단위: mm)

	TRI	DCM	ILTS
거리 오차	953.706	835.682	562.426

더 정확한 분석을 위해 표 4는 총 10만회 수행 결과 실제 이동객체인 receiver와 위치 추적 시스템을 통해 생성된 위치 데이터와의 차를 나타낸다. 표 4를 보면 알 수 있듯이 제안하는 위치 추적 시스템인 ILTS는 기존 위치 추적 시스템보다 receiver의 위치와 더 근사한 위치 추적을 하고 있다.

그림 15는 Device를 최초 작동 시 항상 Kalman Filter의 보정된 위치데이터만을 사용할 경우의 결과를 나타낸다. 그림 15를 보면 알 수 있듯이 안정화 되는 시간까지 큰 오차가 존재함을 볼 수 있다. 그림 16은 ILTS에서 Device의 최초 작동 시 결과를 보여준다. LCM에서 'Filtering\_

Selection Phase'에서 Filtering 사용 여부 판단 알고리즘을 수행하여, 설정 오차 이상의 값이 오차가 유지될 시 Kalman Filter의 보정 데이터를 사용하지 않고, TRI와 DCM의 위치 데이터 중 LCM에서 선택된 위치 데이터를 표현해 주기 때문에 그림 16과 같이 receiver와 더 근사한 위치 데이터를 갖게 되는 것을 볼 수 있다.

## 5. 결론 및 향후 연구

본 논문에서는 정확한 receiver의 위치 추적을 위한 실내 위치 추적 시스템인 ILTS를 제안하였다. ILTS는 TRI와 DCM으로부터 수신된 위치 데이터를 비교 및 분석함으로써 실제 receiver의 위치와 더 근사한 위치데이터를 선택하고, 선택된 위치 데이터를 Kalman Filter를 수행함으로써 보정된 위치 데이터를 얻는 과정을 수행하였다. 이를 통해 receiver와 위치 추적 시스템의 위치 데이터와의 오

차가 현저히 줄어들어 더 정확한 실내 위치 추적을 가능케 하였다.

향후 연구과제로는 생성된 TRI와 DCM의 위치 데이터 중에서 receiver의 위치에 더 근사한 위치 데이터를 선택하기 위해, 급격한 환경 변화가 있음에도 최초 설정된 Selection Rank를 계속적으로 사용하게 되는 문제점을 해결한 환경의 변화에 능동적인 등급화의 차별화 연구가 필요하다.

## 참 고 문 헌

- [1] 박옥선, 정광렬, 김성희, “유비쿼터스 컴퓨팅을 위한 위치인식 기술 및 시스템”, ETRI 주간 기술동향 1098호, 2003
- [2] Xin Wang, Zongxin Wang, “A TOA-Based Location Algorithm Reducing the Errors due to Non-Line-Of-Sight (NLOS) Propagation”, IEEE, pp. 97-100, 2001
- [3] S. Venkatraman, J.Caffery, Jr. H. R. You “Location Using LOS Range Estimation in NLOS Environments”, IEEE, pp. 856-860, 2002
- [4] R. Ertel and J. Reed, “Angle and time of arrival statistics for circular and elliptical scattering models,” IEEE Journal on Selected Areas on Communications, vol 17, pp. 1829-1840, 1999
- [5] Xinrong Li, Kaveh Pahlavan, Jacques Beneat, “PERFORMANCE OF TOA ESTIMATION TECHNIQUES IN INDOOR MULTIPATH CHANNELS”, IEEE, 2002
- [6] M.A. Youssef et al, “A probabilistic clustering-based indoor location determination system”, Technical Report CS-TR-4350 and UMIACS-TR-20002-30, University of Maryland, 2002.
- [7] A. Smailagic et al, “Location sensing and privacy in a context aware computing environment”, Proc. Pervasive Computing, 2001.
- [8] Greg Welch and Gary Bishop, “An Introduction to the Kalman Filter”, UNC-Chapel Hill, TR 95-141, 2004.
- [9] Bao Long Le, Kazi Ahmed, Hiroyuki Tsuji, “Mobile Location Estimator with NLOS Mitigation Using Kalman Filtering”, IEEE, pp.1969-1973, 2003.
- [10] N.J.Thomas, “A Robust Location Estimator Architecture with Biased Kalman Filtering of TOA Data for Wireless Systems”, IEEE 6th, pp. 296-300, 2000.
- [11] Kent K.C. Yu, N.r. Watson, and J.Arrillaga, Fellow, “An Adaptive Kalman Filter for Dynamic Harmonic State Estimation and Harmonic Injection Tracking”, IEEE, vol.20, NO.2, pp.1577-1584, 2005.

◎ 저 자 소개 ◎



**우 성 현 (Sung-Hyun Woo)**

2004년 대전대학교 정보통신공학과 졸업(학사)  
2004년~현재 한밭대학교 정보통신전문대학원 정보통신공학과 재학(석사)  
관심분야 : 공간 데이터베이스, 실내 위치기반 서비스, Kalman Filter, etc.  
E-mail : wsungh123@gmail.com



**전 현 식 (Hyeon-Sig Jeon)**

2005년 한밭대학교 정보통신전문대학원 정보통신공학과 졸업(석사)  
2005년~현재 한밭대학교 정보통신전문대학원 전파공학과 재학(박사)  
관심분야 : 공간 데이터베이스, 실내 위치기반 서비스, etc.  
E-mail : hsjeon@hanbat.ac.kr



**김 기 환 (Ki-Hwan Kim)**

2005년 한양대학교 대학원 전자통신전파공학과(석사)  
2005년~현재 삼성전기 무선통신 솔루션 사업부 전임 연구원  
관심분야 : IEEE 802.15.4a, etc.  
E-mail : kihwan1224.kim@samsung.com



**박 현 주 (Hyun-Ju Park)**

1997년 서울대학교 전산학과 이학박사  
2004년~현재 한밭대학교 정보통신컴퓨터공학부 부교수  
관심분야 : 임베디드 소프트웨어, 공간 데이터베이스, etc.  
E-mail : phj@hanbat.ac.kr