

# 확장된-APIT 테스트 기반의 거리 비종속 위치추정 기법

## (Range-Free Localization Method based on extended-APIT Test)

최 정 욱 <sup>†</sup>                      오 동 익 <sup>\*\*</sup>  
(Jung-Wook Choi)                      (Dong-Ik Oh)

**요 약** 본 논문에서는 제한된 무선 센서 네트워크 영역에서의 물리적인 현상을 사용하여 저 비용으로 객체의 위치를 파악할 수 있는 거리 비종속 위치추정 방식의 대표적인 Approximate Point in Triangle(APIT) 기법에서 발생할 수 있는 위치추정 오차를 줄일 수 있는 확장된-APIT(e-APIT) 기법을 제안한다. 구체적으로는 타겟이 존재하는 영역 추출을 위한 새로운 테스트 방법을 제안하고, 이를 이용해 APIT에서 발생할 수 있는 영역 판단 오류를 최소화 시킨다. 또한 타겟의 주변 환경 정보를 이용하여 추정된 영역을 좁힘으로써 위치추정 오차를 감소시킬 수 있는 방법도 제시한다. 시뮬레이션 결과 본 논문에서 제안하는 확장된-APIT 방식을 사용할 경우 기존의 APIT 방식을 사용할 경우보다 위치 추정의 정확도가 현격히 향상될 수 있는 것으로 파악되었다.

키워드 : 위치추정, 거리 비종속, 확장된-APIT, 무선센서네트워크

**Abstract** In this paper, we propose a range-free localization method that can improve the estimation accuracy of Approximate Point in Triangle(APIT), which is the representative localization method for low cost wireless sensor networks. Specifically, we propose extended-APIT(e-APIT) method, which minimizes the error in deciding whether an object is in an area formed by three beacons. We also propose a way to improve the localization by narrowing down the potential localization area using the signals from neighboring beacons. According to the simulation performed, the proposed e-APIT method demonstrated noticeable accuracy improvement over the conventional APIT method.

Key words : localization, range-free, extended-APIT, wireless sensor network

## 1. 서 론

일반적으로 위치추정은 GPS(Global Positioning System)를 기반으로 하는 실외 위치추정을 의미하였으나, 무선 통신의 발전과 재해 구조를 목적으로 한 미국 연방통신위원회(FCC)의 권고 Docket No. 94-102[1]에 따라 모든 911(응급구조) 요청자의 정밀한 위치를 파악할 필요성이 생기게 되었고, 이후 실외 뿐 아니라 실내에서의 위치추정에 관한 연구도 활성화 되었다[2,3]. 따라서 GPS를 이용할 수 없는 실내와 같은 음영지역에서는 새로운 방식의 측위기술이 필요하며, 이를 위해서는 무선 네트워크 기반에서 다양하고 쉽게 서비스를 확장할 수 있는 위치추정 기술에 대한 연구가 필요하다.

현재 실내외 위치추정 기술은 이동통신, 홈 네트워크, 지능형 로봇, 의료, 텔레메틱스, USN(Ubiquitous Sensor Network) 등 산업 전 분야의 핵심 기술로 활용되고 있

· 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것이며(2009-0071965), 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 중점연구소지원사업으로 수행된 연구임(2010-0020163)

<sup>†</sup> 정 회 원 : 인하대학교 정보전자공통연구소 교수

jwook.choi@inha.ac.kr

<sup>\*\*</sup> 정 회 원 : 순천향대학교 의류IT공학과 교수

dohdoh@sch.ac.kr

논문접수 : 2010년 3월 8일

심사완료 : 2010년 8월 26일

Copyright©2010 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 정보통신 제37권 제6호(2010.12)

으며, 각각의 분야에서 다양한 통신 방식과 측정 기술들이 연구되고 있다[4,5]. 특히 USN 분야에서 위치 기반 서비스의 중요성은 매우 크다. USN 서비스에서는 곳곳에 편재된 센서와 컴퓨터들이 수집한 각종 환경 정보를 효과적으로 공유하고, 이를 기반으로 사용자 및 주변 환경의 상황(context)을 알아내어, 그에 맞는 다양한 서비스를 창출하여 제공하는 상황 인식적(context-awareness) 컴퓨팅의 특성을 갖는다[4]. 이러한 상황 인식 서비스를 가능케 하기 위해서는 사용자 및 사물의 객체인식과 각 객체들의 위치파악이 필수적이다.

본 논문에서는 무선 센서 네트워크에서 저 비용으로 객체의 위치를 파악할 수 있는 거리 비중속 방식의 위치추정 시스템과 이의 위치추정 정확도를 높이기 위한 새로운 솔루션을 제공함을 연구의 목적으로 하고 있다. 이를 위해 기존 거리 비중속 방식의 대표 격인 APIT 위치추정 기법에서 발생하는 추정 오차를 감소시킬 수 있는 확장된-APIT(e-APIT) 위치추정 기법을 제안한다. 구체적으로는 타겟이 포함되는 영역 판단을 위한 e-APIT 테스트를 제안하고, 이를 통해 APIT 테스트에서 발생할 수 있는 부정확한 판단 결과를 제거한다. 또한 타겟의 주변 환경 정보를 이용하여 추정된 영역에서의 위치추정 오차를 감소시킬 수 있는 방법을 제시한다.

이후 논문의 구성은 다음과 같다. 2장에서는 위치추정 기술들을 분류하고 본 논문의 방식인 거리 비중속 기법에 대해 설명한다. 3장에서는 본 논문에서 제안하는 방식의 기반이 되는 APIT 위치추정 기법에 대해 설명하고 APIT 테스트의 판단 오류에 대해 분석한다. 4장에서는 APIT 테스트의 판단 오류를 감소시킬 수 있는 e-APIT 테스트에 대해 설명하고 이를 기반으로 하는 위치추정 기법에 대해 설명한다. 5장에서는 제안하는 e-APIT 위치추정 기법과 기존의 APIT 위치추정 기법의 성능을 시뮬레이션 결과를 통해 분석한다. 6장에서는 연구의 결론을 맺는다.

## 2. 위치추정 기술

### 2.1 위치추정 기술의 분류

위치추정 기술은 크게 삼각측량(triangulation) 방식, 근접(proximity) 방식 및 장면 분석(scene analysis) 방식으로 구분이 가능하다[6].

삼각측량 방식은 객체의 위치를 계산하는데 있어서 삼각형의 기하학적 속성을 사용하며, 측정 거리를 이용하는 레터레이션(lateration)과 측정 각도를 이용하는 앵글레이션(angulation)으로 분류할 수 있다[6]. 레터레이션을 이용하기 위해서는 참조 포인트로부터 객체까지의 거리를 측정할 수 있어야 하는데, 이를 위해서 신호가 전송된 시간 또는 수신 신호의 세기를 거리로 환산한다.

앵글레이션을 이용하기 위해서는 방향성 안테나를 사용하여 수신되는 신호의 각도를 측정한다. 이러한 삼각측량 방식은 다수의 참조 포인트로부터 객체까지의 측정된 거리 또는 각도를 기반으로 위치를 추정하기 때문에 거리 중속(range-based) 방식이라고도 한다.

근접 방식은 객체의 위치를 파악하는데 제한된 영역(range)에서의 물리적인 현상을 사용한다. 즉 알려진 위치 근처에 객체가 존재할 때 그 객체를 인식할 수 있고 이때의 그 알려진 위치를 사용하는 방법이다[6]. 근접 방식에서의 객체 위치 파악 방법은 세 가지로 분류할 수 있다.

첫째는 가장 기본적인 방법으로서 객체의 물리적 접촉을 검출하는 방법이다. 물리적 접촉을 센싱하기 위해서는 압력센서, 터치센서, 전기 용량(capacitive) 검출기 등이 이용된다. 사람의 피부와 직접적인 접촉을 하는 객체들 간의 체내(intra-body) 데이터 통신을 위한 시스템인 Contact[7]가 이러한 물리적 접촉방법을 사용한 대표적 예이다.

둘째, 무선 네트워크에서 모바일 장치가 하나 이상의 AP(Access Point) 범위 내에 있을 때를 모니터링 하는 방법이다. 사무실 환경에서 적외선 셀(cell)을 사용하는 Active Badge Location System[8]과 XeroxParcTAB System[9], 802.11 무선 네트워크를 사용하는 Carnegie Mellon Wireless Andrew[10], 애드혹(Ad-hoc) 네트워크에서의 APIT[11], Centroid[12] 등이 대표적인 예이다.

셋째는, 자동 인식 시스템을 사용하는 방법이다. 신용 카드 단말기, 컴퓨터 로그인 기록, 통화 기록 등과 UPC(Universal Product Code), 상품 코드, 고속도로 카드 시스템, 주입 가능한 가축의 ID 칩과 같은 전자태그[13]를 사용하는 것이 대표적 예이다. 즉 레이블을 스캐닝 하거나, 태그에게 신호를 보내거나, 또는 알려진 위치에서의 트랜잭션을 모니터링 하여 모바일 객체의 위치를 추론하는 방법이다. 이 방식은 영역 혹은 셀 단위 분류에 의한 위치 결정 방법이기 때문에 셀-아이디(Cell-ID) 방식이라고도 한다.

위치추정 기술 중, 장면 분석 방식은 미리 알려진 장면에서 객체나 관찰자의 위치추정을 위해 장면의 특징을 저장해 놓은 후, 객체가 위치한 장면을 저장된 데이터와 비교해 파악하는 방식이다. 관찰된 장면들은 비교하기 쉽도록 특징만을 포함하여 단순화시킨다. 예를 들어 촬영된 장면의 실루엣이나 에지(edge)만 남김으로써 장면을 단순화시킨다. 관찰된 장면의 특징들은 미리 촬영되어 정의된 지도와 같은 데이터 세트에서 검색된다. 반면에, 장면차이(differential) 분석은 추정된 위치의 장면들과 현재 위치의 장면과의 차이점을 찾아낸다. 장면에서 차이가 나는 것은 객체의 움직임에 해당하고, 만약

장면들로부터 특정 포인트를 알게 된다면 객체의 상대적 위치를 파악할 수 있다.

장면 데이터는, 시각적인 이미지, 예를 들어, 웨어러블(wearable) 카메라로부터 획득된 프레임[14]들로 구성할 수 있다. 또는 다른 어떤 측정 가능한 물리적인 현상 예를 들어, 특정 방위나 위치에 객체가 있을 때 발생할 수 있는 전자기적 특징들로도 구성할 수 있다. 마이크로소프트 연구소의 RADAR[15] 위치추정 시스템은 후자의 예이다. RADAR는 건물 전체의 수많은 장소에서 802.11 무선 네트워크 장치의 신호 세기를 측정하여 구성된 데이터 세트를 이용한다[15]. 다른 802.11 네트워크 장치들의 위치는 생성된 데이터 세트의 테이블에 대한 검색을 수행함으로써 계산될 수 있다. 이러한 방식을 지문(fingerprint) 방식이라고 한다[16]. 이 경우 관찰된 특징(신호 세기 값)들을 건물 내에서의 특정 위치와 연관 짓지만, 그 위치들에 대한 기하학적 거리나 각도 등의 측정을 필요로 하지는 않는다.

**2.2 거리 비중속 / 거리 종속 위치추정**

2.1절에서 분류된 세 가지 위치추정 기술은, 측정된 신호 세기나 도착시간 등을 직접적으로 거리로 환산하여 객체의 위치를 추정하는 방식인 거리 종속(range-based) 방식과 신호 세기나 도착시간 등을 직접적으로 이용하지 않고 영역의 포함 여부만으로 객체의 위치를 추정하는 거리 비중속(range-free) 방식으로 크게 구분할 수 있다. 이에 따라, 삼각측량 방식은 거리 종속 방식에 포함되고, 근접 방식은 거리 비중속 방식에 포함된다. 장면 분석 방법은 측정된 신호의 세기를 직접적으로 이용하는 경우 및 이와 유사한 개념을 사용하는 경우이기 때문에 거리 종속 방식에 포함시킬 수 있다.

거리 종속 방식에서는 추가적인 하드웨어를 이용하여 각도 및 거리등을 측정한다. 하지만 위치추정 결과의 정밀도에 비해 거리 종속 방식의 솔루션에서 요구되는 하드웨어의 비용은 매우 높다. 이러한 특징과 문제점들에 대한 대안으로 거리 비중속 방식이 주목받게 되었고, 이 방식을 통해 거리 종속 방식과 견줄 수 있는 수준의 위치추정 결과를 얻고자 하는 노력이 다양하게 이루어지고 있다. 대표적인 위치추정 기법으로는 APIT, DV-hop, Centroid 등이 있다.

**3. APIT 기반 위치추정 기법**

본 장에서는 본 논문에서 제안하는 위치추정 기법의 모체가 되는 APIT 기반의 위치추정 기법에 대해 설명한다.

APIT는 앵커 노드 3개를 선택하고 그 앵커 노드 3개로 구성된 삼각형 안에 타겟 노드가 존재하는지를 판단하는 것을 근간으로 하는 방법이다. 타겟이 다수의 앵커

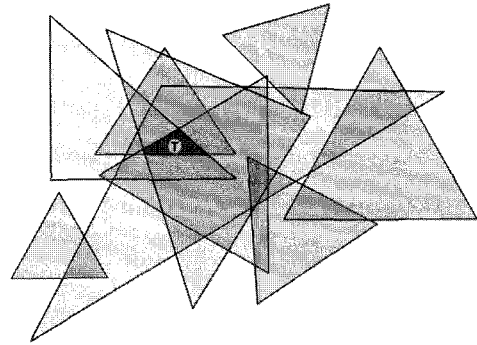


그림 1 APIT 위치추정 기법의 개요

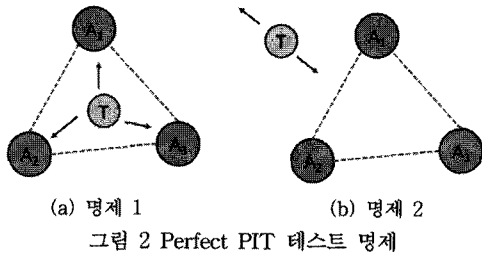
삼각형 조합에서의 내/외부 존재 여부를 알 수 있으면 이 정보를 통합하여 타겟 노드의 위치를 추정할 수 있다. 그림 1은 APIT 위치추정 기법의 전체적인 개념을 설명하는 그림이다.

APIT 기반의 위치추정 기법은 비컨신호 교환, APIT 테스트, 테스트 결과 통합, 무게중심 계산의 네 단계로 진행된다. 이 중 APIT 테스트 단계는 앵커로 구성된 삼각형의 내/외부 어디에 타겟이 존재하는지의 여부를 결정하는 단계로서, 타겟의 위치추정 정확도에 가장 큰 영향을 미치는 단계이다. 하지만 APIT 테스트는 실제 PIT(Point In Triangle) 테스트가 아닌 이를 근사(approximate)하는 방식이기 때문에 삼각형 내/외부 판단에 있어서 오류가 발생할 수 있다.

**3.1 APIT 테스트**

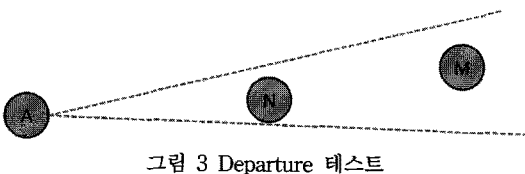
APIT 기반 위치추정 기법의 세 가지 단계 중 APIT 테스트는 비컨 신호가 수신되는 n개의 앵커(이웃앵커) 중에서 중복되지 않도록 3개씩의 앵커를 선택한 뒤, 그 3개의 앵커로 구성되는 삼각형(앵커 삼각형)들을 대상으로 타겟이 내부에 존재하는지 아니면 외부에 존재하는지를 판단하는 과정이다. APIT 테스트는 Perfect PIT 테스트 이론에 기초한다[11]. Perfect PIT 테스트는, “타겟 T가 앵커 A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub>로부터 동시에 멀어지거나 가까워지는 방향이 존재한다면 타겟 T는  $\Delta A_1A_2A_3$ 의 외부에 있고 그렇지 않은 경우에는 내부에 있다”는 이론이다. 이 이론은 다음의 두 명제로부터 유도된 결론이다. 첫째, 만약 타겟 T가  $\Delta A_1A_2A_3$  내부에서 움직인다고 가정하면, 앵커 A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub> 중 적어도 하나의 앵커와는 가까워지거나 멀어진다(그림 2(a)). 둘째, 만약 타겟 T가  $\Delta A_1A_2A_3$  외부에서 움직인다고 가정하면, 앵커 A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub> 모두와 가까워지거나 멀어지는 방향이 존재한다(그림 2(b)).

하지만 이 이론에서 타겟의 움직임이 없다면 내/외부 여부를 판단할 수 없으며, 실사 움직일 수 있더라도, 타겟이 움직일 수 있는 모든 방향을 테스트하는 것은 현



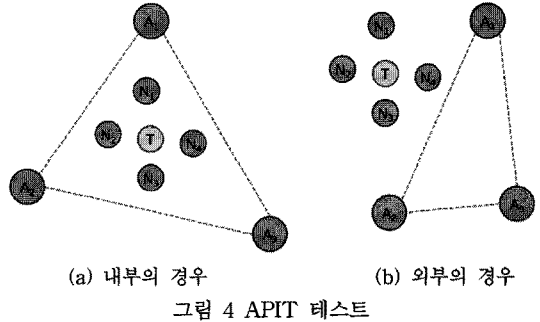
실적으로 불가능하다. 이와 같이 Perfect PIT 테스트 이론이 움직이지 않는 타겟의 위치추정에 있어 현실적인 솔루션을 제공할 수 없기 때문에 Tian He의 기존 연구[11]에서는 이를 근사적으로 접근할 수 있는 APIT 테스트를 제안하고 이를 기반으로 한 위치추정 기법을 제안하였다.

APIT에서는 이동 및 방향 테스트를 위해 이웃노드와의 관계를 이용한다. 이를 위해 기존 연구[11]에서는 신호세기를 비교하는 departure 테스트 결과를 사용한다. Departure 테스트에서는 기준 앵커 혹은 노드로부터 어떠한 노드가 더 멀리 있는지 가까이 있는지를 노드에서 받아들이는 신호의 세기를 비교하여 판단한다. 그림 3의 예에서 보면, 노드 M이 앵커 A로부터 수신한 신호의 세기는 노드 N이 앵커 A로부터 수신한 신호의 세기보다 작기 때문에 노드 M이 노드 N 보다 앵커 A로부터 더 멀리 떨어져 있음을 파악할 수 있다. Departure 테스트의 논리적 타당성은 기존 연구[11]에서 증명되었다.



이를 이용하면 자신(노드)이 실제 이동하지 않고서도 특정 이웃노드의 위치로 이동하는 것을 가정하였을 경우에 앵커와의 거리가 멀어지거나 가까워짐을 파악할 수 있다. 예를 들어, 그림 3에서 노드 N이 이웃노드 M의 위치로 이동한다고 가정한다면, 노드 N의 신호세기는 결국 노드 M의 현재 신호세가 된다. 즉, 신호의 세기가 약해질 것이기 때문에 앵커 A로부터의 거리는 멀어진다고 파악할 수 있다. APIT 테스트는 이러한 방식을 기반으로 수행된다.

APIT 테스트는, “3개의 앵커  $A_1, A_2, A_3$ 에 대해 타겟 T에 비해 모두 가깝거나 모두 먼 이웃노드들이 없다면 T는  $\Delta A_1A_2A_3$  내부에 있다고 판단하고, 그렇지 않은 경우에는 외부에 있다”고 판단하는 이론이다. 그림 4



는 이 이론에 따른 APIT 테스트의 예이다. 그림 4(a)의 경우, 앵커  $A_1, A_2, A_3$ 에 대해 동시에 모두 멀거나 가까운 이웃노드가 없기 때문에 타겟 T는  $\Delta A_1A_2A_3$  내부에 있다고 판단한다. 그림 4(b)의 경우, 이웃노드  $N_2$ 는 앵커 모두에 대해 타겟 T보다 멀고, 이웃노드  $N_4$ 는 앵커 모두에 대해 타겟 T보다 가깝기 때문에 타겟 T는  $\Delta A_1A_2A_3$  외부에 있다고 판단된다.

**3.2 통합 및 무게중심 계산**

모든 이웃앵커들에 대해 APIT 테스트를 수행하고 앵커 삼각형으로 만들어진 각 영역에 대해 판단 결과가 내부일 때는 +1, 외부일 때는 -1의 가중치를 할당한다. APIT 테스트가 모두 종료된 뒤 그림 5처럼 각 삼각형 영역에 할당된 가중치를 전체 영역의 격자에 할당하고 내/외부 판단 결과에 따라 할당된 가중치를 더하는 과정이 통합 단계이다.

APIT 테스트 결과의 통합과정을 마치면 전체 영역에서 가장 높은 가중치를 갖는 영역을 찾을 수 있으며, 이 영역의 무게중심을 구하면 타겟의 추정 위치가 된다. 이 단계가 무게중심 계산의 단계이다.

1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	-1	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	-1	1	1	-1
0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0
0	0	1	1	1	1	1	2	1	0	0	0	0	0	1	1	1	0

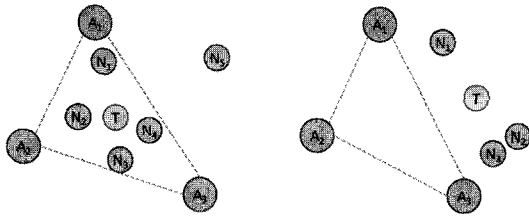
그림 5 APIT 테스트 결과 통합

**3.3 APIT 테스트의 오류**

APIT 테스트에서는 이웃노드를 이용하여 이동 및 방향 테스트를 수행하기 때문에 이웃노드들이 타겟 주변에 배치되어 있는 형태나 이웃노드들의 개수에 따라서 판단의 결과가 달라질 수 있다. 본 절에서는 APIT 테스트를 수행하는데 있어 발생할 수 있는 두 가지 판단 오류에 대해 설명한다.

3.3.1 내부 판단 오류(InToOut 오류)

예를 들어, 그림 6(a)의 상황에서 APIT 테스트를 수행하면 이웃노드  $N_1 \sim N_4$ 는 모두 타겟이 내부에 존재한다고 판단하고 이웃노드  $N_5$ 는 모든 앵커들이 타겟보다 멀기 때문에 타겟이  $\Delta A_1A_2A_3$  외부에 존재한다고 판단한다. 따라서 이웃노드  $N_5$  때문에 타겟 T는 외부에 존재한다고 결론이 내려진다. 이처럼 타겟이 실제로는 앵커 삼각형 내부에 있지만 외부로 판단하는 오류를 InToOut 오류라고 한다.



(a) InToOut 오류 (b) OutToIn 오류  
그림 6 APIT 테스트의 두 가지 오류

3.3.2 외부 판단 오류(OutToIn 오류)

예를 들어, 그림 6(b)의 상황에서 APIT 테스트를 수행하면 이웃노드  $N_1 \sim N_3$ 은 타겟 T보다 앵커를 모두가 멀거나 가깝지 않기 때문에 모두 타겟이 내부에 존재한다고 판단한다. 따라서 타겟 T는 실제로  $\Delta A_1A_2A_3$ 의 외부에 존재함에도 불구하고 외부로 판단하는 이웃노드가 없기 때문에 타겟이 앵커 삼각형 내부에 존재한다고 판단한다. 이처럼 타겟이 실제 앵커 삼각형 외부에 있지만 내부로 판단하는 오류를 OutToIn 오류라 한다.

3.4 APIT 테스트의 두 가지 오류 분석

APIT 테스트는 이웃노드들의 배치 형태, 거리에 따라 판단 오류가 발생할 수 있다.

InToOut 오류가 발생하는 상황은, 타겟으로부터 멀리 떨어져 있는 이웃노드가 포함되어 테스트 할 경우이다. 그림 6(a)와 같이 이웃노드( $N_5$ )가 해당 앵커 삼각형 외부에 걸쳐 있는 상황은 자주 발생할 수 있다. 이러한 이웃노드는 모든 앵커들과의 거리가 타겟보다 멀어지게 되므로 결국 타겟이 삼각형 외부에 존재한다고 판단하게 된다. 즉 타겟의 이웃노드 선택 범위가 커질수록 외부라고 판단할 수 있는 노드가 포함될 가능성이 커지므로, InToOut 오류를 해결하기 위해서는 그 범위를 좁혀 가까운 이웃노드만을 선택하도록 해야 한다. 반면에, 이웃노드가 타겟 주변 어느 방향에 위치하는지는 판단에 큰 영향을 끼치지 않는다. 결국 APIT 테스트의 내부 판단에 있어 중요한 요소는 이웃노드들과의 거리이다.

OutToIn 오류가 발생하는 상황은, 앵커 모두와 동시에 멀어지거나 가까워지는 이웃노드가 없을 경우이다.

예를 들면, 그림 6(b)와 같이 이웃노드( $N_1, N_2, N_3$ )들이 모든 앵커에 대해 타겟보다 가깝거나 멀 수가 없는 위치의 노드들만 있을 경우이다. 그림에서 타겟 T의 좌측이나 우측의 영역에 이웃노드가 있다면 이들은 모든 앵커들과 가깝거나 멀 수 있기 때문에 외부로 판단 할 수 있다. 즉 타겟과 가깝거나 먼 거리와는 상관없이 이웃노드의 배치 형태에 따라 OutToIn 오류가 발생하게 된다. 따라서 OutToIn 오류를 해결하기 위해서는 이웃노드가 고르게 분포될 수 있는 배치 형태가 중요하다. 반면, 이웃노드와의 거리는 판단에 큰 영향을 끼치지 않는다. 결국 APIT 테스트의 외부 판단에 있어 중요한 요소는 이웃노드들의 배치 형태이다.

이렇듯 APIT 테스트의 두 가지 오류를 해결하기 위한 각각의 이웃노드의 조건이 서로 상반되기 때문에 각각의 판단을 보완할 수 있는 추가적인 방법을 이용하여 판단오류를 감소시키는 접근방법이 필요하다.

이러한 오류는 위치추정의 통합 단계에서 타겟을 포함하지 않는 영역에 잘못된 가중치를 부여하게 하여 위치추정의 정확도를 떨어뜨리는 요인으로 작용한다. InToOut 오류들 때문에 가중치를 증가시켜야 할 영역의 가중치가 오히려 감소할 수도 있고, OutToIn 오류까지 포함시키면 가장 높은 가중치를 갖는 영역이 실제 타겟이 위치한 영역과 어긋날 가능성이 있다. 따라서 이러한 오류를 최소화 하는 것이 위치추정의 정확도를 높이기 위한 방법이 될 것이다. 이에 본 연구에서는 앵커 삼각형 내부가 아니기 때문에 외부라고 판단할 것이 아니라, 외부로 판단되었을 경우에 이것이 진짜 외부인지 아닌지를 판단할 수 있는 조건을 제시하고, 이를 통해 보다 정확하게 위치를 추정할 수 있는 방법을 제안하고자 한다.

4. e-APIT 기반의 위치추정 기법

3장에서 살펴본 바와 같이 APIT 위치추정 방식에서는 앵커 삼각형 내/외부에 타겟이 존재하는지를 판단함에 있어 오류가 발생할 수 있고, 이러한 오류가 누적되면 전체적인 위치추정의 정확도가 떨어질 수 있다. 따라서 본 장에서는 APIT 테스트의 판단 오류를 감소시키기 위한 e-APIT 테스트 방식을 제안하고 이를 기반으로 한 위치추정 기법에 대해 설명한다.

4.1 메인 알고리즘

APIT 방식에서의 오류를 감소시키고 위치추정의 정확도를 높이는 것을 핵심으로 하는 e-APIT 방식의 진행은 크게 네 단계로 구분 된다 : 1) 비컨 신호 교환, 2) e-APIT 테스트, 3) e-APIT 테스트 결과 통합, 4) 무게중심 계산. 제안하는 e-APIT 테스트 기반 위치추정 기법의 의사(pseudo) 코드는 다음과 같다.

```

InsideSet = ∅;
OutsideSet = ∅;
For (each triangle  $T_i \in (3 \text{ of } N) \text{ triangles}$ ) {
    If ( $e\text{-APIT}(T_i) = \text{INSIDE}$ )
        InsideSet = InsideSet  $\cup$   $\{T_i\}$ ;
    else if ( $e\text{-APIT}(T_i) = \text{OUTSIDE}$ )
        OutsideSet = OutsideSet  $\cup$   $\{T_i\}$ ;
    else
        continue; // ignore  $T_i$ ;
};
Position = RefinedCOG(InsideSet, OutsideSet);

```

비컨 신호 교환 단계에서는 앵커와 노드들이 비컨 신호를 방송하는 단계로서, 전체 영역의 앵커들 중에서 신호가 수신되는 앵커들을 이웃앵커로 구분하고, 전체 영역의 노드들 중에서 신호가 수신되는 노드들을 이웃노드로 구분하여 e-APIT 테스트에 사용할 이웃앵커와 이웃노드를 결정한다.

e-APIT 테스트 단계에서는 3개의 이웃앵커로 구성된 앵커 삼각형을 대상으로 이웃노드를 이용하여 e-APIT 테스트를 수행한다. 타겟이 앵커 삼각형 내부에 존재하는지 외부에 존재하는지 판단하는 단계이다. 앵커 삼각형은 전체 이웃앵커들 중에서 3개씩 선택하는 방법(조합)으로 모든 앵커 삼각형에 대해 e-APIT 테스트를 수행한다.

e-APIT 테스트 결과 통합 단계에서는 e-APIT 테스트 단계에서 타겟이 앵커 삼각형 내부에 존재하는지, 외부에 존재하는지에 따라 분류된 결과들을 통합한다. 즉 앵커 삼각형을 좌표에 따라 격자에 매핑(mapping) 하고 각 결과 중 내부일 때는 격자에서의 앵커 삼각형 영역을 가중치 1 증가시키고 외부일 때는 가중치 1 감소시킨다.

무계중심 계산 단계에서는 e-APIT 테스트 결과를 통합한 후 실제 타겟의 정보와 비교하여 최고 가중치를 갖는 영역에서 타겟이 존재하지 않는 격자 영역을 분리한다. 이렇게 구해진 최고 가중치 영역에서 무계중심을 구해 타겟의 추정 위치로 결정한다. 이때 위치추정 오차의 보정단계를 추가로 거쳐 위치추정의 정확도를 높인다.

#### 4.2 e-APIT 테스트

APIT 테스트에서는 단 하나의 이웃노드라도 외부라고 판단할 경우 타겟이 외부에 존재한다고 결정하기 때문에 그림 6과 같은 오류들이 발생하게 된다. 따라서 보다 구체적인 판단 조건을 적용하여 오류를 감소시키는 것이 e-APIT의 핵심이다.

실제 앵커 삼각형 외부에 존재하는 타겟을 구분하기 위해 외부에 존재하는 타겟을 살펴보자. 그림 7은 앵커 삼각형 외부에 있는 타겟과 이웃노드의 관계를 나타낸

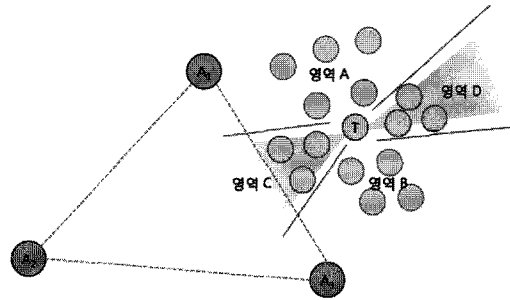


그림 7 앵커 삼각형 외부의 타겟과 이웃노드의 관계

다. 영역 A와 영역 B에 존재하는 이웃노드들은 최소한 하나의 앵커와는 타겟보다 거리가 가깝고 나머지 앵커와는 타겟보다 거리가 멀다. 영역 C에 존재하는 이웃노드 등은 모든 앵커들에 대해 타겟보다 거리가 가깝고, 영역 D에 존재하는 이웃노드들은 모든 앵커들에 대해 타겟보다 거리가 멀다. 명확한 영역의 구분은 불가능하지만, 실제로 타겟이 외부에 존재한다고 하더라도 영역 C 혹은 영역 D에 이웃노드가 존재하지 않는다면 APIT 테스트 결과는 내부가 된다. 그림 6(b)의 이웃노드들은 모두 외부라고 판단할 수 없는 그림 7의 영역 A와 영역 B에 존재한다. 즉 단 하나의 노드도 앵커들과의 거리가 타겟보다 모두 멀거나 모두 가까운 경우가 생기지 않는다. 이 상황에서는 결국 내부라고 판단할 수밖에 없는 OutToIn 오류가 발생하지만, 거리 비중속 방식에서 이러한 오류를 해결할 수 있는 방법을 찾기는 어려워 보인다. 이에 본 연구에서는 OutToIn 오류는 고려하지 않고 또 다른 판단 오류인 InToOut 오류를 줄이는 방법으로 위치추정의 오차를 감소시키는 방법을 제시하고자 한다.

실제 외부에 있는 타겟일 경우 일반적으로 타겟의 주위를 이웃노드들이 고르게 둘러싸고 있다고 가정할 때, 모든 앵커와의 거리가 타겟보다 가깝거나(그림 7의 영역 C) 모든 앵커의 거리가 타겟보다 먼(그림 7의 영역 D) 이웃노드가 동시에 존재 한다면 타겟은 확실히 앵커 삼각형 외부에 있다고 결정할 수 있다. 왜냐하면 내부에 존재하는 타겟일 경우 모든 앵커에 대해 가깝거나 먼 이웃노드는 존재할 수 없기 때문이다. e-APIT 테스트에서는 이러한 특성을 이용하여 새로운 판단 기준을 제안하려고 한다.

e-APIT 테스트는 기존의 APIT 테스트에 의해 앵커 삼각형의 외부로 판단된 것 중, 실제로 삼각형 외부에 있는지의 여부를 확인하는 새로운 알고리즘을 적용하여, InToOut 오류의 가능성을 줄이는 방법이다. 또한, 이렇게 판단된 결과들 중 내부 혹은 외부라고 명확히 판단할 수 없는 불명확한 결과들을 제거함으로써 테스트 결과

통합 단계에서 보다 명확한 결과들을 이용하도록 한다.

본 논문에서 제안하는 e-APIT 테스트는, “3개의 앵커  $A_1, A_2, A_3$ 에 대해 타겟  $T$ 에 비해 모두 가깝거나 모두 먼 이웃노드들이 없다면  $T$ 는  $\Delta A_1A_2A_3$  내부에 있다고 판단하고, 그렇지 않은 경우 중 타겟  $T$ 에 비해 모두 가까운 이웃노드와 모두 먼 이웃노드가 동시에 존재해야만  $T$ 는  $\Delta A_1A_2A_3$  외부에 있다고 판단한다. 그리고 내부 혹은 외부로 결정되지 못한 결과는 불명확한 결과로 판단한다”는 알고리즘이다.

이 알고리즘은 타겟이 외부에 존재한다고 판단되었을 경우, 그림 7의 C와 D영역에 노드가 존재하면 타겟이 확실히 삼각형 외부에 있다고 판단될 수 있다. 그렇지 않은 경우에는 타겟이 삼각형 외부에 있다는 것을 확신할 수 없는 경우이기 때문에 테스트 결과 통합 단계에서 가중치 벡터(weight vector)값을 가/감해 주지 않음으로써 위치추정 오차를 감소시킬 수 있다. 이 알고리즘을 적용한 e-APIT 테스트에서는 그림 6(a)와 같은 InToOut 오류를 제거할 수 있게 된다.

**4.3 e-APIT 테스트 결과 통합**

모든 앵커 삼각형에 대해 e-APIT 테스트가 완료되면 타겟의 위치를 추정하기 위해 그 결과를 통합한다. 타겟이 내부에 존재한다고 판단한 앵커 삼각형의 영역은 가중치 1을 할당 하고 외부에 존재한다고 판단한 앵커 삼각형의 영역은 가중치 -1을 할당 한다. 4.2절에서 설명한 불명확한 판단의 경우에는 가중치를 부여하지 않는다. 이 과정의 결과를 통해 가장 높은 가중치를 갖는 영역 즉, 타겟이 존재할 수 있는 영역이 도출된다. 이 과정은 APIT 테스트 결과 통합과 유사하지만 보다 명확한 내/외부 결과만을 사용하는 차이가 있다. 이의 의사(pseudo) 코드는 다음과 같다.

```

For (each triangle  $T_i \in (3 \text{ of } N) \text{ triangles}$ ) {
    If ( $e\text{-APIT}(T_i) == \text{INSIDE}$ )
        AddPositiveValue( $T_i$ );
    Else If ( $e\text{-APIT}(T_i) == \text{OUTSIDE}$ )
        AddNegativeValue( $T_i$ );
    Else
        IgnoreTriangle( $T_i$ );
};
Find the areas with Max value;
    
```

**4.4 무게중심 계산(Refined COG)**

e-APIT 테스트의 결과 통합 과정을 마치면 가장 가중치가 높은 영역이 도출되며 이 영역이 바로 타겟이 존재한다고 판단되는 영역이다. 타겟의 추정 위치는 그 영역의 무게중심(COG : Center Of Gravity)을 구해 결

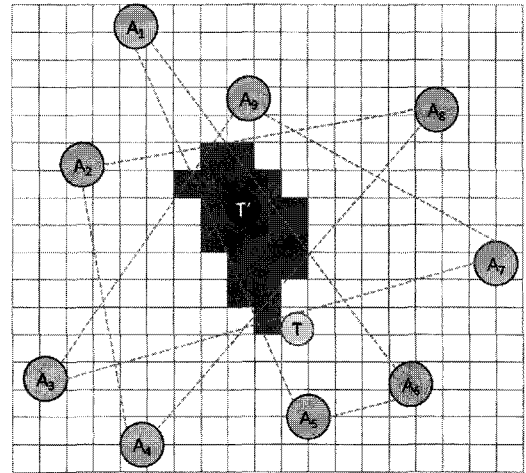


그림 8 추정된 영역에서의 무게중심 계산

정하는데, 이때 APIT 방식과는 다르게 e-APIT 방식에서는 추정영역을 좁혀나가는 과정을 추가로 수행한다. 그림 8은 APIT 방식에서 사용하는 무게중심 계산 방법으로서 최고 가중치 영역에서의 무게중심을 계산하여 타겟의 위치를 도출해 낸다.

**4.4.1 위치추정 오차 발생원인**

가장 높은 가중치를 갖는 영역이 타겟을 포함하는 영역이라고 판단하고 단순히 그 영역의 무게중심을 구하여 타겟의 위치를 결정하게 될 때 위치추정의 오차가 발생할 수 있다. APIT 기반의 위치추정 기법에서 사용하는 이 방식은, 영역 전체에서 무게중심을 구하기 때문에 추정된 영역의 크기가 작을 경우에만 작은 오차가 발생한다. 하지만 테스트의 오류들을 포함하여 추정 영역이 커졌을 때에는 타겟이 무게중심 근처에 있어야만 작은 오차가 발생하고, 그렇지 않은 경우(무게중심에서 타겟이 벗어나 있는 경우)에는 필요 이상의 위치추정 오차가 발생할 수 있다.

**4.4.2 위치추정 오차 보정 방법**

최고 가중치를 갖는 영역이라고 하더라도 타겟을 포함하지 않거나 타겟이 중심에 있지 않다면 추정 위치 오차는 커지게 되므로 이를 보완할 수 있는 방법이 필요하다. 따라서 본 연구에서는 최고 가중치 영역을 도출한 후에, 추정 위치를 결정함에 있어 타겟 주변 앵커의 정보를 이용하여 위치추정 오차를 감소시키는 방법을 제안한다.

타겟의 위치추정을 수행하기 위해 이용하는 정보는 비컨신호가 수신되는 이웃앵커와 이웃노드들이다. 이들의 departure 테스트 결과를 이용하여 가장 먼 앵커와 가장 가까운 앵커 정보를 찾을 수 있다. 이 정보를 이용하면 추정된 영역에서 타겟이 존재할 수 있을 가능성이

있는 영역을 추출할 수 있다. 구체적으로 타겟의 실제 가장 가까운 앵커와 가장 먼 앵커의 정보를 추정된 최고 가중치 영역의 각 격자에서 가장 가까운 앵커와 가장 먼 앵커의 정보와 비교했을 때, 그 정보가 다르다면 해당 격자는 실제 타겟이 존재할 수 없는 영역이므로 이를 제외할 수 있다. 이로써 추정된 최고 가중치 영역에서 실제 타겟이 존재할 수 있는 영역을 추가적으로 좁혀나갈 수 있기 때문에 단순히 무게중심을 계산하는 방법보다 적은 오차로 위치추정이 가능하다. 이 과정은 아래의 의사(pseudo) 코드를 따른다.

```

FAncor; //The farthest anchor from target
CAncor; //The closest anchor from target
For (each square  $S_i \in$ TheHighestWeightedSquare){
    If( (farthestAnchor( $S_i$ ) != FAncor) ||
        (closestAnchor( $S_i$ ) != CAncor) )
        Remove  $S_i$ ;
};
Find Center of Gravity;
    
```

예로서 그림 8의 상황에서 실제 타겟의 위치는 T 위치이며 이 때 가장 가까운 앵커는  $A_5$ 이고 가장 먼 앵커는  $A_1$ 이다. e-APIT 테스트를 수행하고 결과 통합을 통해 추정된 영역의 무게중심을 구하여 타겟의 좌표를 구했을 때의 결과가 T'이라고 가정하자. 이 때 가장 가까운 앵커는  $A_9$ 이고 가장 먼 앵커는  $A_4$ 이다. T'과 T의 가장 가까운, 가장 먼 앵커 정보를 비교하면 서로 다르다. 따라서 최고 가중치를 갖는 영역의 격자들을 대상으로 비교하여 가장 가까운 앵커와 가장 먼 앵커가 T의 정보와 동일한 영역을 다시 검색하고 그 영역에서의 무게중심을 구한다. 최고 가중치를 갖는 영역에서도 동일한 거리관계를 찾지 못할 경우에는 APIT 방식에서와 마찬가지로 최고 가중치를 가진 영역의 무게중심으로 추정 위치를 결정한다. 그림 9는 그림 8의 위치추정 결과를 보정하는 예이며, T''를 구하여 이를 추정 위치로 결정함으로써 위치추정 오차를 감소시킬 수 있다.

**5. e-APIT 기반의 위치추정 시뮬레이션**

본 논문에서 제안하는 e-APIT 기반의 위치추정 시스템과 기존의 APIT 기반의 위치추정 시스템의 성능을 비교 평가하기 위하여 타겟의 위치를 추정하는 시뮬레이션 프로그램을 개발하였다. 성능(위치추정 정확도)에 영향을 미칠 수 있는 수신 신호 세기의 오차, 앵커의 신호 세기(앵커 인식 범위), 노드의 신호 세기(노드 인식 범위), 앵커의 개수, 노드의 개수 등을 파라미터 화하여 다양한 상황에서의 성능평가가 가능하도록 하였다.

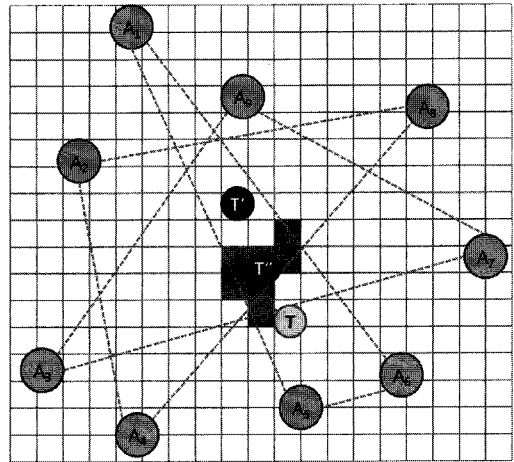


그림 9 위치추정 결과 보정의 예

**5.1 위치추정 오차**

본 시뮬레이션에서는 정사각형으로 정의된 테스트 영역 내에서 타겟의 위치를 추정한다. 사각형의 테스트 영역 즉, 좌측 최상단 좌표와 우측 최하단 좌표까지의 범위에서 노드들이 위치할 수 있기 때문에 발생할 수 최대 오차는, 좌표 (0,0)에서부터 위치할 수 있는 최대 좌표(테스트 영역의 최대 크기)까지의 거리이며, 오차는 타겟의 실제 좌표와 추정된 좌표까지의 거리를 최대 거리로 나눈 비율로 정의한다. 오차 E를 식으로 나타내면 다음과 같다.

$$E = \frac{\sqrt{(TR_x - TE_x)^2 + (TR_y - TE_y)^2}}{\sqrt{(Area_x)^2 + (Area_y)^2}} \quad (1)$$

여기서,

$TR_x$  : 타겟의 실제 x좌표

$TR_y$  : 타겟의 실제 y좌표

$TE_x$  : 타겟의 추정된 x좌표

$TE_y$  : 타겟의 추정된 y좌표

$Area_x$  : 사각형 테스트 영역의 최대 x좌표

$Area_y$  : 사각형 테스트 영역의 최대 y좌표이다.

**5.2 신뢰도 및 정확도**

'위치추정 시스템 X는 95%의 신뢰도 내에서 90%의 정확도를 나타낸다'라고 했을 경우, 위치추정 시스템 X는 평균 95% 범위 내에서는 평균 10% 이내의 오차가 발생한다는 것을 의미한다[6]. 이를 통해 위치추정 시스템의 성능을 평가할 수 있다. 이러한 신뢰도와 정확도를 제공하기 위한 기준으로서, 본 논문에서는 각 시뮬레이션에서 실패율을 계산하여 신뢰도  $R_{conf}$ 를 제공하고, 성공한 경우에 정확도  $R_{acc}$ 를 제공하도록 하였다.

신뢰도는 전체 위치추정 시도 횟수와 위치추정 실패



횃수의 차를 전체 위치추정 시도 횃수로 나누어 표현할 수 있으며, 위치추정 실패 횃수는 3개미만의 앵커가 선택되어 위치추정이 불가능한 경우와 3개 이상의 앵커가 선택되었음에도 불구하고 판단 결과가 모두 외부인 경우의 두 가지를 포함한다. APIT 및 e-APIT 방식의 위치추정 기법은 최소 3개의 앵커가 존재해야지만 위치추정이 가능하기 때문에 3개미만의 앵커가 선택된 경우는 위치추정이 불가능하다. 또한 이 기법들은 이웃앵커들로 구성된 영역 내에 타겟이 존재하는 위치를 추정하는 방법이므로, 그 이웃앵커들로 구성된 삼각형 조합 모두 외부라고 판단된 경우에는 위치추정이 가능한 영역 내에서 타겟이 어디에 존재하는지 알 수 없는 경우이므로 위치추정이 불가능하다. 따라서 신뢰도  $R_{conf}$ 는 식 (2)와 같이 계산될 수 있다.

$$R_{conf} = \frac{Cnt_T - (Cnt_2 + Cnt_{allOut})}{Cnt_T} \quad (2)$$

여기서,  $Cnt_T$ 는 전체 위치추정 시도 횃수,  $Cnt_2$ 는 3개미만의 앵커가 선택된 횃수,  $Cnt_{allOut}$ 은 모두 외부로만 판단한 위치추정 횃수이다.

정확도  $R_{acc}$ 는 성공한 전체 시뮬레이션에서의 위치추정 오차 평균으로 표현할 수 있으며, 다음과 같이 계산될 수 있다.

$$R_{acc} = \left( \sum_{i=0}^{n-1} E_i \right) / n \quad (3)$$

여기서,  $E$ 는 5.1절에서 설명한 위치추정 오차이며,  $n$ 은 전체 시뮬레이션 횃수이다.

### 5.3 시뮬레이션 과정

시뮬레이션을 위해 정의한 파라미터 중 수신 신호 세기의 오차는, 신호 세기 감쇠 및 지연 등의 효과를 반영하기 위해 신호세기의 20% 범위 내에서 비주기적 오차를 발생시키고(장애물 등의 간섭으로 인한 신호 감쇠를 표현) 신호의 5% 범위 내에서 주기적 오차(실제 환경에서 발생할 수 있는 신호간섭 등에 의한 일반적 오차)를 발생시켰다. 이 수치는 기존 연구[11]에서 제시한 신호 모델의 오차를 유추하여 결정한 값이다. 또한 앵커의 신호 세기(앵커 인식 범위), 노드의 신호 세기(노드 인식 범위), 앵커의 개수, 노드의 개수 등을 변화시키며 위치추정 오차를 및 위치추정 가능 비율 등에 관한 시뮬레이션을 진행하였다.

본 시뮬레이션에서는 400×400 크기의 영역에서 중복되지 않도록 노드와 앵커의 좌표를 랜덤하게 생성하였으며, 결과를 통합하고 추정 영역을 도출하기 위한 격자의 크기는 4로 설정하였다. 랜덤한 타겟의 좌표를 생성한 뒤 e-APIT와 APIT 두 가지 방식의 위치추정 기법을 시뮬레이션 하였다.

### 5.4 시뮬레이션 결과

e-APIT 방식과 APIT 방식의 성능 분석을 위해 이웃노드의 밀도, 이웃앵커의 밀도 등을 변화시키며 시뮬레이션을 수행하였다. 이웃노드는 전체 영역의 노드 중에서 테스트에 이용되는 타겟과 근접한 노드들이며, 이웃앵커는 전체 영역의 앵커 중에서 테스트에 이용되는 타겟과 근접한 앵커들이다.

#### 5.4.1 이웃노드의 밀도(개수와 인식 범위)

전체 영역 중에서 노드의 신호 전파 거리에 따라 통신 가능한 영역이 설정되며 이에 따라 이웃노드의 개수가 달라진다. 이때 타겟이 위치추정에 사용하는 이웃노드의 개수를 이웃노드의 밀도라 한다. 테스트에 사용되는 이웃노드의 밀도는 이웃노드 인식 범위와 관련이 있다. 이웃앵커의 범위가 고정되어 있는 상황에서 이웃노드의 인식 범위를 늘리면, 테스트에 이용하는 이웃노드의 개수가 증가하므로 결국 이웃노드의 밀도가 증가하게 된다. 따라서 먼저 전체 테스트 영역에서 노드의 개수를 늘려 이웃노드의 밀도를 증가시키는 방법으로 시뮬레이션을 수행 하였다. 그림 10은 전체 영역의 노드 개수를 100에서 1000까지 증가시켜, 이웃노드의 밀도를 변화시킨 시뮬레이션의 위치추정 오차를 나타낸다. 전체 테스트 영역에서 노드의 개수가 증가함에 따라 타겟이 위치추정에 이용할 이웃노드의 개수가 증가(노드 밀도 증가)하며, 이웃노드의 밀도가 증가함에 따라 e-APIT 기반의 위치추정 기법은 오차가 감소하고 APIT 기반의 위치추정 기법은 오차가 증가함을 알 수 있다.

또 다른 방식으로는 노드의 인식 범위를 변화시킴으로써 이웃노드의 밀도를 증가시키는 방법으로 시뮬레이션을 수행하였다. 그림 11은 노드의 인식 범위 변화에 따른 위치추정 오차에 관한 결과이다. 이 방법에서도 노드의 인식 범위가 증가함에 따라 이웃노드의 밀도가 증가하게 되며, 이웃노드 밀도의 증가에 따라 e-APIT 기반의 위치추정 기법은 오차가 감소하고 APIT 기반의 위치추정 기법은 오차가 증가함을 확인할 수 있다.

위 결과의 원인을 파악하기 위해 시뮬레이션 데이터에서 e-APIT와 APIT 테스트의 결과들을 분석하였다.

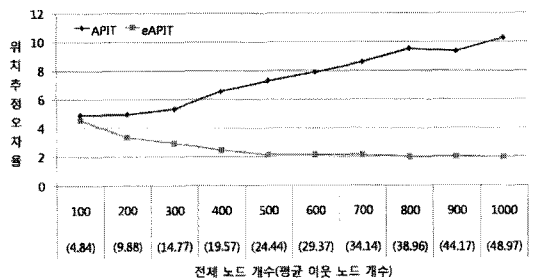


그림 10 노드의 개수에 따른 위치추정 오차

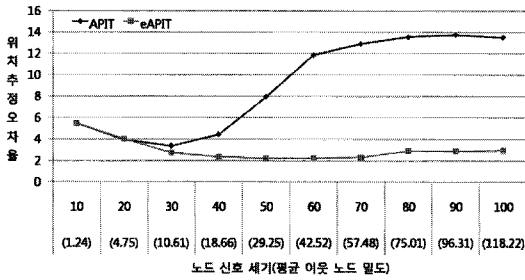


그림 11 노드의 인식 범위에 따른 위치추정 오차

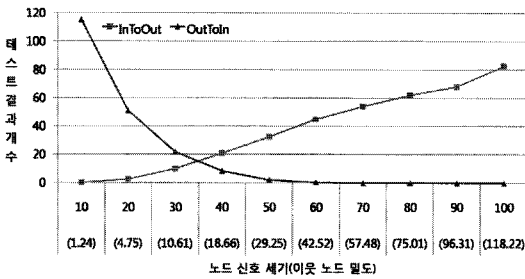


그림 12 노드 밀도에 따른 APIT 테스트의 판단 결과 개수

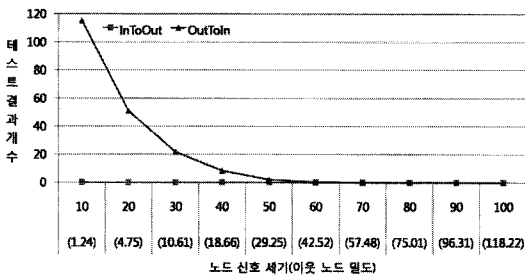


그림 13 노드 밀도에 따른 e-APIT 테스트의 판단 결과 개수

그림 12는 APIT 방식의 시뮬레이션 결과이고, 그림 13은 e-APIT 방식의 시뮬레이션 결과이다. 이 두 그래프를 통해, 노드의 밀도가 증가함에 따라 e-APIT 기반의 위치추정 기법은 위치추정 오차가 감소하고, APIT 기반의 위치추정 기법은 위치추정 오차가 증가하는 원인을 파악할 수 있다. 두 그래프에서 InToOut은 잘못된 내부 판단, OutToIn은 잘못된 외부 판단을 의미한다.

내부 판단(OutToIn)은 APIT 테스트와 e-APIT 테스트 모두 동일한 조건으로 판단하기 때문에 같은 결과를 나타낸다. 그러나 기존의 APIT 테스트와 보다 엄격한 외부 판단 조건을 적용한 e-APIT 테스트가 확연히 구분되는 점은 외부 판단(InToOut) 결과이다. e-APIT 테스트에서 InToOut 판단은 불명확한 판단으로 무시하기 때문에 전혀 발생하지 않지만, APIT 테스트에서는 이웃노드 밀도에 비례하여 증가하는 것을 확인할 수 있다.

내부 판단 결과는 영역에 가중치를 증가시켜 최고 가

중치 영역을 만들어내는 역할이고, 외부 판단 결과는 영역의 가중치를 감소시켜 제외할 영역을 만들어내는 즉, 최고 가중치 영역을 축소시키는 역할을 수행한다. 따라서 위치추정을 위해서는 정확한 내부 판단 결과가 무엇보다도 중요한데, InToOut 오류는 결과 통합 단계에서 가중치를 증가시켜야 하지만 오히려 감소시키기 때문에 위치추정의 정확도에 큰 영향을 끼친다. 이러한 이유 때문에 e-APIT 테스트 기반의 위치추정 기법이 더 적은 위치추정 오차를 나타냄을 알 수 있다.

5.4.2 이웃앵커의 밀도(개수와 인식 범위)

전체 영역 중에서 앵커의 신호 전파 거리에 따라 통신 가능한 영역이 설정되며 이에 따라 이웃앵커의 개수가 달라진다. 이렇게 타겟이 위치추정에 사용하는 이웃앵커의 개수를 이웃앵커의 밀도라 한다. 이웃노드의 밀도와 마찬가지로 이웃노드의 범위가 고정되어 있는 상황에서 이웃앵커의 인식 범위를 늘리는 방법은, 테스트에 이용하는 이웃앵커의 개수를 증가시키므로 결국 이웃앵커의 밀도를 증가시킨다. 이를 테스트하기 위해 전체 테스트 영역에서 앵커의 개수를 늘려 테스트에 이용하는 방법으로 시뮬레이션을 수행 하였다. 그림 14는 전체 영역의 앵커의 개수를 10에서 100범위 내에서 증가시켜, 이웃앵커의 밀도를 변화시킨 시뮬레이션의 위치추정 오차를 나타낸다. 전체 테스트 영역에서 앵커의 개수가 증가함에 따라 타겟이 위치추정에 이용할 이웃앵커의 개수가 증가(앵커 밀도 증가)하며, 이웃앵커의 밀도가 증가함에 따라 e-APIT 방식과 APIT 방식의 위치추정 기법 모두 위치추정 오차가 감소하지만 e-APIT 방식의 성능이 더 우수함을 확인할 수 있다.

또 다른 방법인 앵커의 인식 범위를 변화시킴으로써 테스트의 이용하는 이웃앵커의 밀도를 증가시키는 방법으로 시뮬레이션을 수행하였다. 그림 15는 앵커의 인식 범위 변화에 따른 위치추정 오차에 관한 결과이다. 이 결과 역시 앵커의 인식 범위를 증가시킴으로써 이웃앵커의 밀도가 증가하게 되며, 앵커의 인식 범위 증가에 따라 두 방식의 위치추정 오차가 감소하지만 e-APIT 방식의 성능이 더 좋은 것을 확인할 수 있다. 이는 이웃

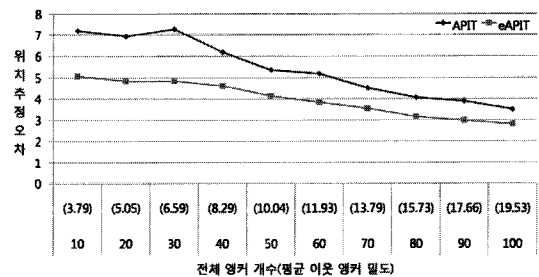


그림 14 앵커 개수에 따른 위치추정 오차

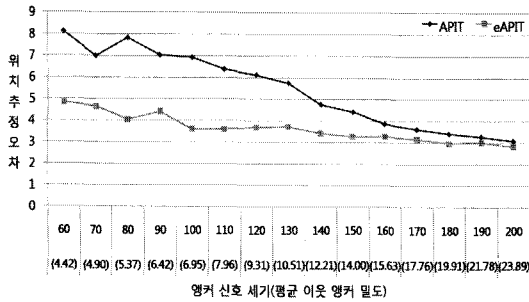


그림 15 앵커의 인식 범위에 따른 위치추정 오차

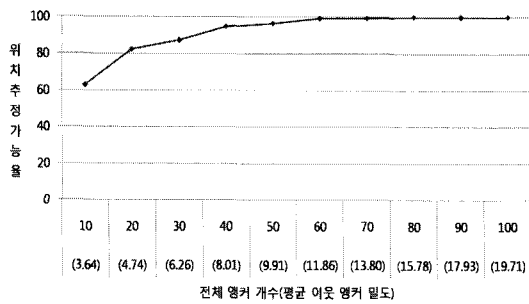


그림 16 앵커 밀도에 따른 위치추정 가능성

노드의 인식 범위는 변함없기 때문에 앵커의 인식 범위가 증가함에 따라 타겟을 포함할 수 있는 더 큰 앵커 삼각형들이 생성되게 되고, 따라서 위치를 추정하기 위해 필요한 내부라고 판단되는 영역이 증가하기 때문이다.

e-APIT 및 APIT 위치추정은 앵커 기반의 시스템이기 때문에 두 가지 모두 앵커의 밀도가 증가하면 위치추정 오차가 감소한다. 앵커의 밀도가 증가할수록 테스트할 삼각형의 개수가 증가하고 형태가 다양해짐에 따라 정확한 판단 결과의 개수도 증가하기 때문에 위치추정 정확도가 향상됨을 확인할 수 있다. 같은 앵커 밀도일지라도 e-APIT 방식이 APIT 방식보다 더 적은 추정 오차가 발생하는 이유는 5.4.1항에서 설명한바 같이 e-APIT 방식은 InToOut 오류가 발생하지 않기 때문이다.

그림 16은 앵커의 밀도 변화에 따라 위치추정 가능 비율(신뢰도)을 나타내는 그래프이다. APIT 및 e-APIT 테스트는 앵커가 3개 미만일 경우 테스트할 수 없기 때문에 위치추정 자체가 불가능하다. 따라서 앵커의 밀도에 따라 위치추정이 가능한 비율을 판단하여 최소 기대할 수 있는 신뢰도를 얻기 위한 기준을 설정할 수 있다.

이 그래프에 따라 400x400의 영역에서 100개의 노드, 120의 앵커의 인식 범위, 50의 노드 인식 범위인 테스트 환경일 경우, 최소 90% 이상의 신뢰도(추정 가능 비율)를 얻기 위해서는, 전체 영역에 30개 이상의 앵커를 배치하여야 한다는 것을 알 수 있다.

### 5.4.3 이웃앵커와 이웃노드의 밀도에 따른 위치추정 정확도

위에서 설명한 이웃앵커의 밀도 변화 및 이웃노드의 밀도 변화에 따른 시뮬레이션의 종합적인 결과는 그림 17과 그림 18에서 확인할 수 있다.

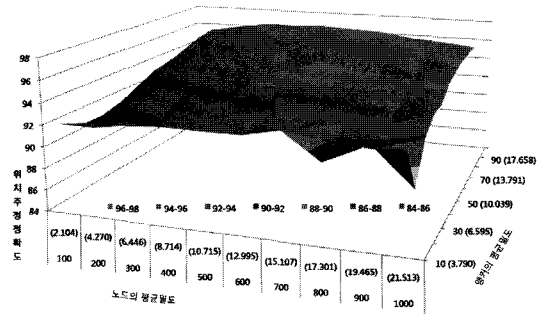


그림 17 노드/앵커 밀도에 따른 위치추정 정확도(APIT)

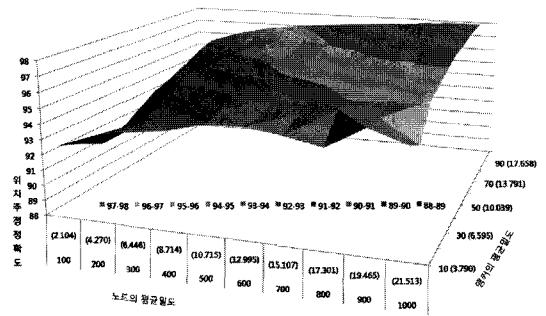


그림 18 노드/앵커 밀도에 따른 위치추정 정확도(e-APIT)

APIT 기반 위치추정 기법의 정확도는 앵커의 밀도에 비례하고 노드의 밀도에 반비례한다. e-APIT 기반 위치추정 기법의 정확도는 앵커의 밀도와 노드의 밀도에 비례하며 APIT 기반 위치추정 기법과 비교하여 훨씬 적은 오차가 발생함을 확인할 수 있다. 결론적으로 평균 위치추정 정확도에 있어 e-APIT 기반의 위치추정 기법이 APIT 기반의 위치추정 기법에 비해 성능 향상을 가져왔음을 확인할 수 있다. 또한 90% 신뢰도 내에서 95% 이상의 정확도를 나타내기 위해서는, APIT 기반의 위치추정 기법의 경우 앵커의 개수가 최소 70개 이상이 필요하지만 e-APIT 기반의 위치추정 기법의 경우는 최소 40개의 앵커만으로도 충분함을 확인할 수 있다.

### 5.4.4 위치추정 결과

그림 19는 4.4절에서 설명한 위치추정 결과 보정의 예이다. e-APIT 방식으로 추정된 영역에서 단순한 무게중심을 구하지 않고 주변 앵커 정보를 활용하여 타겟이 존재할 가능성 있는 영역을 축소시킨 뒤 무게중심을

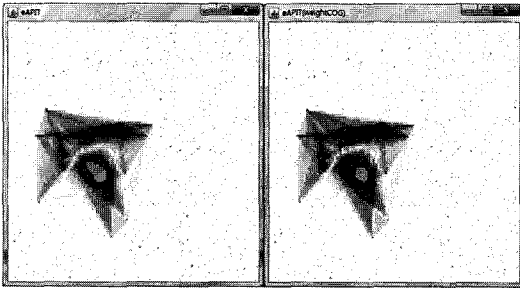


그림 19 위치추정 결과 보정의 예

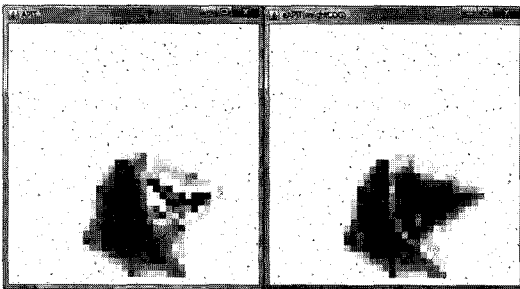


그림 20 APIT 방식과 e-APIT 방식의 위치추정 결과 예

구한 결과를 나타낸다. 테스트 환경은 앵커 50, 노드 800, 앵커의 신호세기 120, 노드의 신호세기 50, 격자 크기 6이며, 17개의 이웃앵커와 46개의 이웃노드가 위치추정에 사용되었다.

그림 19에서 자홍색(magenta)의 점은 앵커를, 청록색(cyan)의 점은 노드를 나타내며, 흑색(black) 영역은 타겟이 존재하지 않는(가중치가 0 미만) 영역, 청색(blue) 영역은 타겟이 존재하는(가중치가 0 초과) 영역을 나타낸다. 그리고 적색(red) 영역은 타겟이 존재한다고 추정되는(최고 가중치) 영역을 나타낸다. 이 결과 기존의 단순한 무게중심 계산방법(그림의 좌측)보다 4.4절에서 제안한 보정 방법(그림의 우측)이 타겟을 포함하는 훨씬 작은 영역(그림에서 적색 영역)을 도출함을 확인할 수 있다.

그림 20은 앵커의 개수 70, 노드의 개수 400, 앵커의 신호세기 120, 노드의 신호세기 40, 격자 크기 10의 테스트 환경에서 APIT 방식과 e-APIT 방식으로 타겟의 위치를 추정한 결과이다.

위의 위치추정 시뮬레이션에서는, 22개의 이웃앵커와 13개의 이웃노드가 위치추정에 사용되었으며, APIT 방식의 위치추정 결과(그림의 좌측)와 비교하여 e-APIT 방식(그림의 우측)의 위치추정이 보다 적은 오차로 위치추정함을 확인할 수 있다.

## 6. 결론

본 논문에서는 대표적인 거리 비중속 위치추정 방식

중의 하나인 APIT 기반의 위치추정 기법에서 발생하는 추정 오차를 감소시키기 위해 e-APIT 기반 위치추정 기법을 제안하였다. e-APIT 테스트에서는 APIT 테스트가 앵커 삼각형 외부에 있는 것으로 판단한 타겟을 다시 한 번 테스트하여, 실제로 삼각형 외부에 있는지의 여부를 확인할 수 있도록 하는 새로운 알고리즘을 제시하였다. 이 테스트를 통해 APIT 테스트에서 매우 빈번히 발생하는 InToOut 판단 오류를 제거시킬 수 있었는데, 그 결과 타겟을 포함하는 삼각영역에 대한 신뢰도를 높여 추정 위치의 오차를 감소시킬 수 있었다. 또한 타겟이 존재할 가능성이 있는 영역이 도출된 후, 이웃앵커를 사용한 정제의 과정을 거쳐 이 영역을 축소시킴으로써 보다 높은 정확도를 갖는 위치추정이 이루어질 수 있도록 하였다. 결과적으로 e-APIT 위치추정은, 기존 APIT 위치추정에서 InToOut 오류를 제거하고, 정제된 무게중심 계산 방법을 사용함으로써 위치추정의 정확도를 현격히 향상시킬 수 있었다. 시뮬레이션 결과 e-APIT 기반의 위치추정 기법이 평균 앵커의 개수가 40개 이상일 때, 90% 신뢰도 내에서 96% 이상의 정확도를 가지는 것으로 평가 되었다.

e-APIT 테스트에서는 APIT 테스트 판단 결과에 대해 추가적인 비교를 수행하는 과정이 필요하므로, 위치추정을 위한 연산량이 증가할 수 있다. 또한 신호의 세기를 기반으로 하는 거리 비중속 방식 모두에게서 나타날 수 있는 무선 신호의 불규칙성으로 인한 오차가 발생할 수 있다. 따라서 향후, e-APIT 테스트의 정확한 판단 및 연산량 감소, 수신 신호 세기의 오차 등을 해결할 수 있는 연구가 추가적으로 필요하다.

## 참고 문헌

- [1] Public Notice, CC Docket No. 94-102, "Wireless Telecommunications Bureau Provides Guidance On Carrier Reports On Implementation Of Wireless E911 Phase II Automatic Location Identification," *Federal Communications Commission*, Apr. 2000.
- [2] H.S. Ahn, J.Y. Lee, W.P. Yu, K.S. Han, "Indoor Localization Technique for Intelligent Robotic Space," *Electronical and Telecommunication Trends analysis*, vol.22, no.2, pp.48-57, Apr. 2007. (in Korean)
- [3] Y.S. Cho, S.Y. Cho, B.D. Kim, S.H. Lee, J.C. Kim, W.S. Choi, "Technical Trend of Indoor/Outdoor Seamless Positioning," *Electronical and Telecommunication Trends analysis*, vol.22, no.3, pp.22-28, Jun. 2007. (in Korean)
- [4] S.H. Kim, J.H. Kim, J.S. Bae, D.S. Kim, "Ubiquitous Network based Service Technologies," *Magazine of Institute of Electronics Engineers of Korea*, vol.30, no.11, pp.89-101, Nov. 2003. (in Korean)

[5] J.H. Park, M.K. Kim, J.H. Paik, "Industrial Structure Analysis and Market Development Strategy Direction of Location Based Service," *Information and Communications Magazine*, vol.20, no.4, pp.92-102, April 2003 (in Korean)

[6] J. Hightower, G. Borriello, "Location Sensing Techniques," *IEEE Computer*, vol.34, no.8, pp.57-66, Aug. 2001.

[7] K. Partridge, L. Arnstein, G. Borriello, T. Whitted, "Fast intrabody signaling," *Demonstration at Wireless and Mobile Computer Systems and Applications*, Dec. 2000.

[8] R. Want, A. Hopper, V. Falcao, and J. Gibbons., "The Active Badge Location System," *ACM Transactions on Information Systems*, vol.40, no.1, pp.91-102, Jan. 1992.

[9] R. Want, B. Schilit, N. Adams, R. Gold, K. Petersen, J. Ellis, D. Goldberg and M. Weiser, "The PARCTAB ubiquitous computing experiment," *Technical Report CSL-95-1, Xerox Palo Alto Research Center*, Mar. 1995.

[10] A. Hills, "Wireless Andrew," *IEEE Spectrum*, vol.36, no.6, pp.49-53, Jun. 1999.

[11] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," *Proc. of the 9th Annual International Conference on Mobile Computing and Networking*, pp.81-95, Sep. 2003.

[12] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications Magazine*, vol.7, no.5, pp.28-34, Oct. 2000.

[13] R. Want, D.M. Russell, "Ubiquitous Electronic Tagging," *IEEE Distributed Systems Online*, vol.1, no.2, Feb. 2000.

[14] T. Starner, B. Schiele, and A. Pentland. "Visual context awareness via wearable computing," *Proc. of the International Symposium on Wearable Computers*, pp.50-57, Oct. 1998.

[15] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.3, pp.775-784, Mar. 2000.

[16] K.W. Kolodziej, J. Hjelm, "Local Positioning Systems: LBS Applications and Services," CRC press, 488, 2006.

학교 정보전자공동연구소 연구교수. 관심분야는 무선센서네트워크, 유비쿼터스 컴퓨팅, 임베디드 시스템, 운영체제, 위치추정 알고리즘, 인체통신 등



오 동 익

1985년 The City University of New York 전산학과(학사). 1989년 Florida State University 전산학과(석사). 1997년 Florida State University 전산학과(박사). 1997년~2006년 순천향대학교 컴퓨터학부 부교수. 2003년~2004년 교육부 국제교육진흥원 컴퓨터교과연수단 단장. 2007년~현재 순천향대학교 의료IT공학과 교수. 관심분야는 u-헬스케어 시스템, RFID, 유비쿼터스 컴퓨팅, 임베디드 시스템, 운영체제, 실시간 시스템, 프로그래밍언어 등



최 정 옥

2003년 순천향대학교 컴퓨터학부(학사)  
2005년 순천향대학교 전산학과(석사). 2009년 순천향대학교 컴퓨터학과(박사). 2009년~2010년 이화여자대학교 컴퓨터공학과 박사후연구원. 2010년~현재 인하대