

ESP32 기반 CSI 데이터를 활용한 위치 분포 시각화

임민형, 류재영, 조영진 정연호[‡]

부경대학교

limminhyeong@gmail.com, {jeyeongryu, heo5810}@naver.com, [‡] yhchung@pknu.ac.kr

Location Distribution Visualization Based on CSI Data from ESP32 Devices

Min Hyeong Lim, Jae Yeong Ryu, Young Jin Cho, Yeon Ho Chung[‡]

Dept of Information and Communications Engineering., Pukyong National University

요 약

본 연구에서는 저비용 ESP32 보드를 활용하여 무선 신호의 Channel State Information(CSI) 데이터를 수집하고 이를 기반으로 단말의 위치를 추정하였다. 세 개의 ESP32 장치를 삼각형으로 배치하고, 각 장치로부터 수신된 CSI magnitude 값을 가중치로 사용하는 weighted centroid 기법을 통해 시간에 따른 단말의 위치를 추정하였다. 이후 Kernel Density Estimation(KDE)을 적용하여 추정된 위치 분포를 공간 히트맵 형태로 시각화함으로써, 단말의 이동 경로와 체류 위치를 직관적으로 파악할 수 있도록 하였다. 본 방식은 복잡하거나 고가의 측위 시스템 없이도 CSI 데이터를 활용한 실내 위치 추정 및 시각화가 가능함을 보여준다.

I. 서 론

GPS가 실내 환경에서는 정확한 위치 정보를 제공하지 못하기 때문에, 실내 위치 추정을 위해 Wi-Fi, Bluetooth, UWB(Ultra-Wideband) 등 다양한 무선 신호를 활용한 다양한 기술 기반의 위치 추정 시스템이 제안되고 있다.[1] 이 중에서도 Wi-Fi는 기존 인프라를 그대로 활용할 수 있다는 점에서 접근성이 높고, 추가적인 하드웨어 설치가 필요 없다는 장점을 가진다. Wi-Fi 기반 위치 추정 기법 중에서는 주로 RSSI(Received Signal Strength Indicator)를 이용한 방식이 널리 사용되어 왔으나, RSSI는 환경 변화나 잡음에 민감하여 정밀한 위치 추정에는 한계가 있다. 이에 따라, 최근에는 물리 계층 정보인 Channel State Information(CSI)을 활용한 연구가 주목받고 있다.[2] CSI는 송수신 채널의 주파수 응답 정보를 서브캐리어 단위로 제공하며, RSSI에 비해 훨씬 더 세밀하고 정교한 정보를 제공할 수 있다. 본 연구에서는 저비용 IoT 장치인 ESP32 보드를 활용하여 CSI 데이터를 수집하고, 이를 기반으로 단말의 위치를 시간적으로 추정한 후, Kernel Density Estimation(KDE)을 통해 공간 분포를 시각화하는 방식을 제안한다. 세 개의 ESP32 장치를 삼각형 형태로 배치하고, 각 장치로부터 수신된 CSI magnitude를 가중치로 활용한 weighted centroid 방식을 통해 단말의 위치를 추정한다. 이후 추정된 위치 이력을 KDE로 처리하여 히트맵을 생성함으로써, 단말의 이동 경로와 체류 지역을 직관적으로 파악할 수 있는 시각화를 제공한다. 이러한 방식은 복잡한 학습 기반 모델이나 고가의 장비 없이도, 상대적으로 간단한 계산을 통해 실시간에 가까운 수준의 위치 추정 및 시각화가 가능하다는 장점을 가진다. 본 논문에서는 제안하는 방식의 설계, 구현 과정 및 시각화 결과를 제시하고, 실험을 통해 그 가능성을 검증한다.

고정되었으며, 각 장치의 위치는 실험 공간 내에서 (0, 0), (1, $\sqrt{3}$), (2, 0)으로 설정하였다. 스마트폰 단말은 실내 공간에서 자유롭게 이동하였고, 각 ESP 보드는 스마트폰이 송신한 Wi-Fi 패킷을 수신하여 CSI 데이터를 획득하였다. 데이터는 USB 직렬 통신을 통해 PC에 저장되었으며, 수집 주기는 약 15~50ms로 확인되었다.

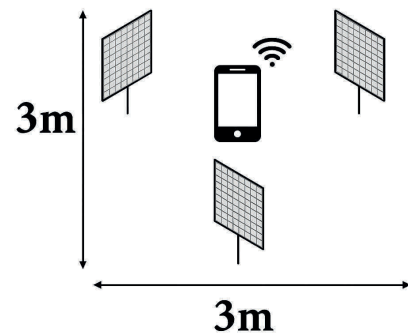


그림1. 실험환경

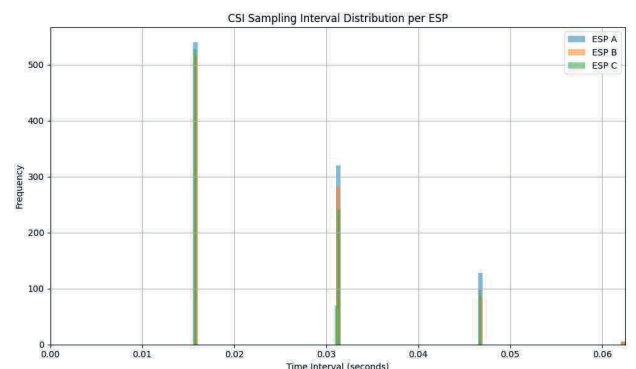


그림2. 수집값 주기 히스토그램

II. 본론

가. 실험 환경 구성

본 연구에서는 ESP32 칩이 탑재된 세 개의 Wi-Fi 송수신 보드를 활용하여 CSI 데이터를 수집하였다. 세 ESP 장치는 삼각형 형태로

나. CSI 데이터 수집 및 전처리

ESP32의 CSI 수집 기능은 Espressif에서 제공하는 CSI 드라이버를 기반으로 활성화되었으며, 수신된 CSI는 서브캐리어별로 실수와 허수값이 번갈아 나열된 정수 배열 형태로 전송되었다. 데이터 전처리 과정은 Python 스크립트를 통해 실시간으로 수행되었으며, 주요 단계는 다음과 같다:

1. 문자열 형태로 수신된 CSI 데이터를 정수 배열로 파싱
2. 실수/허수 성분 중 실수 성분만 추출하여 CSI 벡터를 구성
3. 벡터의 절댓값 평균을 계산하여 ESP 장치별 CSI magnitude로 변환
4. 타임스탬프 및 장치 태그(A, B, C)와 함께 저장하여, 시간 순서에 따른 신호 세기 분석이 가능하도록 구성

이 과정을 통해 시간 흐름에 따른 각 ESP 장치의 수신 세기 정보를 연속적으로 확보할 수 있었으며, 이후 위치 추정 및 시각화에 활용되었다.

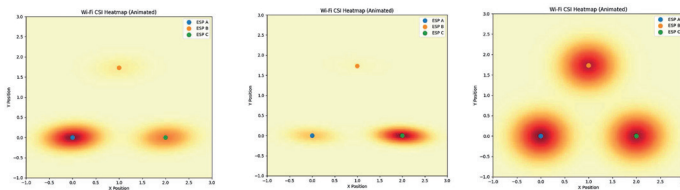
다. 위치 추정 방법

위치 추정에는 weighted centroid localization 기법을 적용하였다. 각 시간 프레임에서 ESP 장치가 수신한 magnitude 값을 정규화하여 가중치로 사용하고, 각 ESP의 위치에 이를 곱하여 단말의 추정 위치를 계산하였다. 수식은 다음과 같다:

$$\vec{P}_{est} = \sum_{i=1}^3 w_i \cdot \vec{p}_i, \quad w_i = \frac{m_i}{\sum_{j=1}^3 m_j}$$

여기서 \vec{p}_i 는 ESP i 의 좌표, m_i 는 해당 ESP가 수신한 CSI magnitude, w_i 는 정규화된 가중치이다.

4. 프레임별 위치 분포 시각화



(a) 프레임 50 (b) 프레임 100 (c) 프레임 150

III. 결론

본 연구에서는 저비용 ESP32 장치를 활용하여 CSI 데이터를 수집하고, 이를 기반으로 단말의 위치를 추정한 후 KDE 기법을 통해 위치 분포를 시각화하는 시스템을 구현하였다. 복잡한 지도 매핑이나 학습 기반 모델 없이도 CSI magnitude와 weighted centroid 방식을 통해 단말의 이동 경로 및 체류 위치를 직관적으로 파악할 수 있었으며, 이는 실내 위치 기반 서비스에 유용하게 적용될 수 있을 것으로 기대된다. 또한, CSI의 위상 정보를 포함한 고차원 벡터 특성을 고려할 때, 향후에는 t-SNE, PCA, UMAP 등의 차원 축소 기법을 통해 CSI 벡터 간의 상대적 거리 구조를 학습하고 시각화하는 채널 차팅(Channel Charting) 방식으로 확장할 수

있다. 이를 통해 단순한 위치 분포 분석을 넘어, 물리적 좌표 없이도 단말 간 공간 관계를 파악하고 무선 환경 변화 감지나 경로 예측 등 다양한 응용에 활용할 수 있을 것으로 기대되며, 향후에는 보다 정밀한 모델링과 다양한 채널 환경에서의 실험을 통해 시스템의 일반성과 실효성을 높이는 방향으로 연구를 진행할 예정이다.

ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (2023R1A2C2006860).

참 고 문 헌

- [1] Zafari, F., Gkelias, A., & Leung, K. K. (2019). "A Survey of Indoor Localization Systems and Technologies." *IEEE Communications Surveys & Tutorials*, 21(3), 2568 - 2599.
- [2] C. Studer, S. Medjkouh, E. Gonultas, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47682 - 47698, Aug. 2018, doi: 10.1109/ACCESS.2018.2866979.