

TDoA 확률화 이미지를 활용한

딥러닝 기반 타겟 추적 모델

Deep Learning Model for Robust Target Tracking using TDoA Probabilistic Image

요 약

UWB (Ultra Wide Band) 기반 실내 측위 방식 중 TDoA (Time Difference of Arrival)를 활용한 방식은 범용적으로 사용되는 방식이지만, AWGN (Additive White Gaussian Noise)이 많은 환경일 때 위치 추적 성능이 현저히 떨어진다. TDoA의 AWGN을 제거하는 방법이 많이 연구되어 왔지만 해당 방식은 별도의 측위 방법을 필요로 하며 다양한 환경에 대한 강건성(Robustness)이 부족하다. 본 논문에서는 AWGN이 포함된 TDoA를 그대로 사용해 물체의 위치를 강건하게 추정하는 ‘TDoA probabilistic image based target tracking (TPITT)’ 방법을 제안한다. TPITT는 TDoA를 통해 각 영역에 물체가 존재할 확률을 이미지화하고, 'Convolution-LSTM' 모델을 통해 물체의 좌표를 추정한다. 실험을 통해 제안하는 방법이 다양한 환경에서 강건하고 낮은 예측 오차를 보임을 입증하였다. 특히, AWGN이 많은 환경에서 사전 연구 ‘TDoA image based target tracking (TITT)’에 비해 제안 방법이 더 효과적임을 확인하였다.

ABSTRACT

"Ultra Wide Band (UWB) based indoor tracking methods utilizing Time Difference of Arrival (TDoA) are commonly used; however, their performance significantly degrades in environments with high Additive White Gaussian Noise (AWGN). Although many studies have attempted to remove AWGN from TDoA, these approaches require additional tracking methods and lack robustness in various environments. In this paper, we propose a 'TDoA Probabilistic Image Based Target Tracking (TPITT)' method that robustly estimates an object's location using TDoA with AWGN. TPITT generates probability images of object presence in each region using TDoA and employs a 'Convolution-LSTM' model to estimate object coordinates. Experiments demonstrate the proposed method's robustness and low prediction error across diverse environments. Notably, TPITT is more effective than the prior study 'TDoA Image Based Target Tracking (TITT)' in environments with high AWGN."

키워드 : UWB 통신시스템; TDoA; 실내 위치 측위; 실시간 측위; 딥러닝;

Key Words : UWB communication systems; TDoA; Indoor tracking; Real time localization; Deep learning;

I. 서 론

위치추적 기술은 물체나 사용자의 위치를 파악하기 위해 널리 사용되며, 이때 널리 사용되는 기술은 글로벌 네비게이션 위성 시스템 (Global Navigation Satellite System)이다^[1]. 하지만 GNSS 기술의 큰 단점 중 하나는 실내에서의 위치 확인이 쉽지 않다는 것이다^[2]. 이러한 어려움을 해결하고자 Wi-Fi와 Bluetooth, UWB (Ultra Wide Band) 등 대체 기술들을 개발해 실내에서의 정확한 위치를 파악하는 연구가 진행되고 있다.

그 중 UWB는 500MHz 이상의 넓은 주파수 대역을 사용하여 짧은 시간 동안 낮은 출력의 펄스 신호로 100m 이내에서 데이터를 송수신하는 무선통신 기술이다^[3]. UWB는 높은 측위 정확성, 다중 경로 간섭에 대한 저항성, 측정값에 대한 안정성, 물리 계층(PHY layer)에서 난수를 활용한 보안성과 같은 장점을 가진다. 하지만 UWB 기술은 2000년 초반 당시 경쟁 기술이었던 Wi-Fi에 비해 미흡한 산업 생태계(ecosystem)와 하드웨어 기술 부족으로 인해 주목받지 못했다^[4]. 시간이 흘러 칩 기술의 발달로 현재 UWB는 앞선 장점 덕분에 재 주목을 받았으며 현재 다양한 분야에서 연구 및 활용되고 있다^[5].

UWB를 활용해 무선기기 간의 거리를 측정하는 방식은 크게 time of arrival (ToA)과 two way ranging (TWR)이 있다. 그 중, ToA는 송신기에서 빛을 쏘아 수신기에서 반사되어 돌아오는 시간을 측정한 뒤, 빛의 속도를 곱하여 거리를 구하는 방식이다. 하지만 ToA 방식은 태그(송신기)의 주소 바인딩이 필요해 통신 가능한 기기의 개수가 한정되어 있으며, 장치 간 여러 번의 통신을 하기에 배터리 소모가 크다^[6]. 문제를 보완하기 위해 서로 다른 앵커(수신기)에서 측정된 신호 시간의 차이를 이용하는 TDoA (Time Difference of Arrival) 방식이 주로 사용된다. TDoA를 활용한 위치 추적 방식에서 태그는 앵커와 개별적인 통신을 하지 않고 블링크 메시지(Blink Message)만을 전송함으로써 배터리 소모를 절감할 수 있다. 또한, 주소 바인딩이 없고 한 번의 전송으로 많은 태그에 신호를 보낼 수 있어, 동시에 다수의 태그와 통신이 가능하다.

그러나 TDoA 방식은 복잡한 비선형 방정식

해를 찾는데 많은 컴퓨터 자원을 소모해야 한다는 단점을 가진다. TDoA 비선형 방정식의 계산 복잡성을 해결하고자 Chan, Taylor, Kalman Filter 등을 활용한 최적화 알고리즘이 제안되었다^[6-7]. 해당 방법들로 인해 TDoA 기반 위치 추정 속도는 개선되었지만, 입력 값의 오차에 민감하게 반응하기 때문에^[15], 장애물이 많은 공간에서는 사용하기 어려운 방법이다.

신호 오류에 강건한 물체 추적 모델을 구축하기 위해, 머신러닝 및 딥러닝 기법을 활용한 방법들이 연구되고 있다^[8-13]. 많은 방법론들은 모델을 활용해 AWGN이 제거된 순수한 신호를 획득하는 것을 목표로 한다. 하지만 이런 방식들은 신호 보정 과정이 복잡하며 실험 환경 변화에 따른 성능이 달라진다는 한계가 존재한다.

본 논문에서는 AWGN이 많은 공간에서 측정된 UWB 신호를 그대로 활용해 두 앵커에 대한 TDoA 쌍곡선 방정식 이미지로 변환한 뒤, 딥러닝을 활용해 물체의 위치를 추정하는 모델을 제안한다. 제안된 방법은 이미지 변환 과정을 통해 AWGN이 측위에 미치는 영향을 감소시킨다. 게다가, 측정된 TDoA를 그대로 사용하는 종단간(End-to-End) 방식이기에 프로세스가 간단하므로 실시간 위치 측정에 적합하다. 마지막으로, 연속적으로 움직이는 타겟의 특성을 고려해 시계열 모델을 도입함으로써 성능을 크게 향상시켰다. 이러한 개선점을 통해 TPITT는 다양한 상황에서 범용적으로 사용할 수 있는 방식임을 실험을 통해 증명하였다.

II. 본 론

2.1 Related Works

2.1.1 Indoor tracking using deep learning

TDoA 측위 시스템의 일반적인 작동 방식은 1) UWB 태그가 메시지를 송신, 2) 주변의 UWB 앵커들이 과정 1에서 보낸 메시지를 수신, 3) 중앙 컴퓨팅 유닛(CCU)은 전체 UWB 앵커들의 메시지 송수신 로그를 수집하여 각 앵커들 간 신호 도달시간 차이(TDoA)를 계산, 4) TDoA를 이용해 쌍곡선 방정식 도출, 5) 방정식들의 해를 계산하여 UWB 태그의 위치 추정^[3]과 같다. 하지만 이러한 방식은 비선형 방정식들의 해를 풀어야 하

기에 AWGN이 많이 발생하는 공간에서 얻은 TDoA를 사용할 경우 전혀 다른 쌍곡선 방정식이 생성되어 결과값이 현저하게 나빠지는 문제점을 지니고 있다^[15].

이러한 문제점을 해결하고자 현재 딥러닝을 활용한 측위 시스템이 연구되고 있다. 딥러닝은 컴퓨터 시뮬레이션 또는 실제 데이터 수집을 통해 얻은 대량의 학습 데이터를 활용하여 복잡한 패턴과 관계를 자동으로 인식하는 인공지능 알고리즘의 한 분야이다. 딥러닝 기반 TDoA 측위 기술은^[9-13] 높은 측위 정확성, AWGN 패턴 학습에 따른 강건성, 그리고 수동적 특성 설계 없이 자동으로 유용한 특성을 학습하는 능력을 가지고 있어 현재 많은 연구가 진행되고 있는 분야이다.

노이즈에 강인한 딥러닝 모델을 학습하는 방법에는 신호의 연속성을 활용한 시계열 기반의 방법이 존재한다. 예를 들어, GCC-PHAT (Generalized Cross Correlation-Phase Transform)을 사용해 steering vector로 분리한 UWB 신호를 BLSTM (Bi-Directional LSTM)을 이용해 AWGN이 제거된 TDoA를 구하는 연구가 있다^[9]. 또한, UWB 신호에서 cross correlation을 이용한 관측 값에서 신호들 간 분리 작업을 통해 TDoA를 구한 뒤, LSTM (Long Short Term Memory) 모델을 사용하여 태그의 위치를 구하는 연구도 존재한다^[10]. 그리고 UWB 신호에 포함된 AWGN을 제거하기 위해 학습된 LSTM 모델을 이용해 정확한 TDoA 값을 찾은 후, weighted KNN (K-Nearest Neighbor) 모델로 타겟의 위치 좌표를 찾아내는 연구 또한 존재한다^[11].

이 밖에도 신호 분야에서 자주 사용되는 컨볼루션 연산을 활용한 CNN (convolutional neural network) 모델 또한 자주 사용되고 있다. 예를 들어 연구는^[12] CIR (channel impulse response) 값을 입력 데이터로 하여 물체의 위치를 찾아내는 CNN 모델을 개발한 연구이다. 또한, 웨이블릿 변환 (Wavelet Transform)을 통해 신호들을 이미지로 변환한 뒤, CNN 모델을 통해 TDOA 값을 찾는 연구 또한 존재한다^[13].

딥러닝을 사용한 선행 연구들은 정확한 TDOA를 얻기 위해 노력했지만, 보정을 위해 여러 연산과 파이프 라인을 거치기에 실시간으로 실내에서 위치를 추적하는 모델로 적합하지 않다. 또

한, 환경이 복잡할수록 신호 변형은 기하급수적으로 늘어나기에, 물체의 움직임과 관측 환경을 모두 고려해서 정확한 TDOA를 보정할 수 있는 알고리즘 구현은 매우 어렵다.

해당 문제를 보완하고자 TDoA image based target tracking (TITT) 모델이 등장하였다^[16]. TITT 모델은 매우 복잡한 실내 환경에서 타겟의 정확한 위치 추적을 진행할 때 좋은 성능을 보이는 모델이다. TITT 모델에서는 TDoA를 이미지로 변환하여 UWB 신호에 대한 AWGN의 영향력을 줄이고, 이를 CNN 모델에 넣어 타겟의 위치 좌표를 추적하는 과정을 수행한다. 실험 결과, TITT는 높은 AWGN의 상황에서 다른 모델에 비해 강건한 위치 추정 결과를 보였다.

그러나 TITT는 AWGN이 많이 발생하는 극단적인 환경에서만 실험된 한정적인 결과를 보여 한계가 있다. 또한, TDoA 기반 방법에서는 신호의 연속성을 고려한 접근이^[9-11] 좋은 성능을 보이지만, TITT는 시계열 특성을 고려하지 않았다. 이에 더해, TDoA 이미지 픽셀들은 TDoA 쌍곡선 방정식의 추정 결과이지만, TITT를 통해 구한 추정 값이 확률적(stochastic)이 아닌 결정론적(deterministic)으로 표현되고 있다. 불확실성 고려하지 않는 TITT의 결정론적 방식은 훈련 데이터에 대한 과적합 위험과 강건한 모델 학습의 어려움에 대한 한계를^[20] 지니고 있다.

2.1.2 Convolutional LSTM

LSTM은 RNN (Recurrent Neural Network) 모델 중 한가지이며 Vanilla RNN의 최대 취약점인 장기 의존성 문제 (the problem of Long-Term Dependencies)와 기울기 손실 현상 (Vanishing gradient problem)을 해결한 모델이다. 이미지 데이터를 LSTM 모델에 사용하기 위해선 이미지를 벡터로 변환해야 하는데, 이 과정에서 이미지의 지역적 정보 (topological information)가 손실되어 학습 시간과 성능이 크게 저하된다.

Convolutional-LSTM은 앞서 진술한 단점을 보완하여 발전된 LSTM 모델이다^[17]. Convolutional-LSTM은 기존의 LSTM 모델과 상당히 유사하지만 1) 해당 모델은 입력-망각-출력 게이트 (i,f,o)와 셀 입력, 출력 상태 (X,C,H)가 3차원 벡터로 계산된다. 이를 통해 3차원 채널 데이터의 손실

없이 시퀀스 데이터의 학습이 가능하다. 2) LSTM에서 행렬 곱이 사용되는 부분은 모두 합성 곱으로 대체되었다 라는 2가지의 차이점을 가지고 있다. 2 가지 차이점 덕분에 시간적, 공간적 상관성 (Spatiotemporal correlation)을 동시에 학습할 수 있다는 큰 장점이 있다. 그 결과 컴퓨터 비전 분야에서 LSTM 모델보다 우수한 성능을 보였다^[22].

2.2 Proposed UWB Indoor Tracking Approach

2.2.1 Model Structure Overview

본 절에서는 이번 논문에서 제안하는 모델의 구조를 설명한다. 제안하는 TPITT (TDoA Probabilistic Image based moving-Target Tracking) 모델은 Figure 1과 같으며 4개의 UWB 앵커와 1개의 태그를 사용해 얻은 TDoA를 입력 값으로 한다. 모델의 프로세스는 다음과 같다: 1) 관측한 TDoA는 격자 지역에 태그가 존재할 확률로 구성된 이미지로 변환하는 과정을 거친다. 이렇게 변환한 이미지들은 특정 시간 t에 각각의 앵커에서 관측한 TDoA 3D Tensor로 합쳐진다. 2) TDoA 3D Tensor는 시계열 데이터로 전환하기 위해 고정된 창 (windows)을 이용해 4차원 구조의 데이터(TDoA 4D Tensor)로 전환한다. 3) 타겟의 위치 정보가 적혀 있는 이미지 데이터 (TDoA 4D Tensor)를 Convolutional-LSTM 신경망 모델을 사용하여 이

동하는 태그의 위치 좌표를 추적한다. 각 단계에 대한 설명은 앞으로 이어지는 단락에서 보다 자세히 진행한다.

2.2.2 Conversion of TDoA to Image

신호를 이미지로 변환하는 과정은 TDoA 값을 이미지로 변환하여 UWB 신호에 포함되어 있던 AWGN의 영향을 낮추는 작업을 수행한다. 이미지로 변환하는 과정은 Figure 2와 같으며, 과정을 수행하기 위해선 앵커 2개의 공간 좌표와 TDoA 값, 이미지 크기, 격자 크기, 최소 거리 차이 상수 (ϵ), temperature (T)를 표현하는 하이퍼파라미터가 필요하다. 이때, 격자 크기는 주어진 공간을 얼마나 자세하게 나눌 것인지 결정하는 변수이다. 또한, 최소 거리 차이 상수는 앵커와 격자 중점의 거리 차이와 TDoA와 유사하다고 비교할 때 필요한 작은 양수이며 temperature는 시그모이드 (sigmoid) 함수의 출력값을 조절해주는 양수이다^[18].

신호를 이미지로 변환하는 과정 1단계는 격자 중점과 앵커 간의 TDoA 행렬을 구하는 것이다. 앵커의 좌표를 A라 정의하면 격자의 중점 좌표 C는 Equation 1과 같이 계산되며 앵커 2개 (anchor_ref & anchor_n)를 통해 가로와 세로가 이미지 크기인 행렬 M1, M2가 만들어진다.

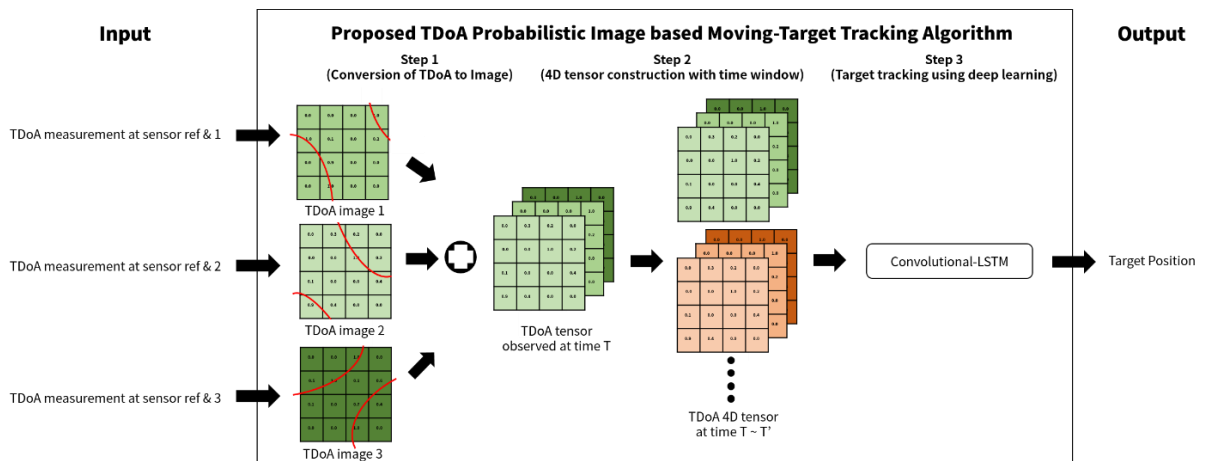


Fig. 1. Overview of the proposed method

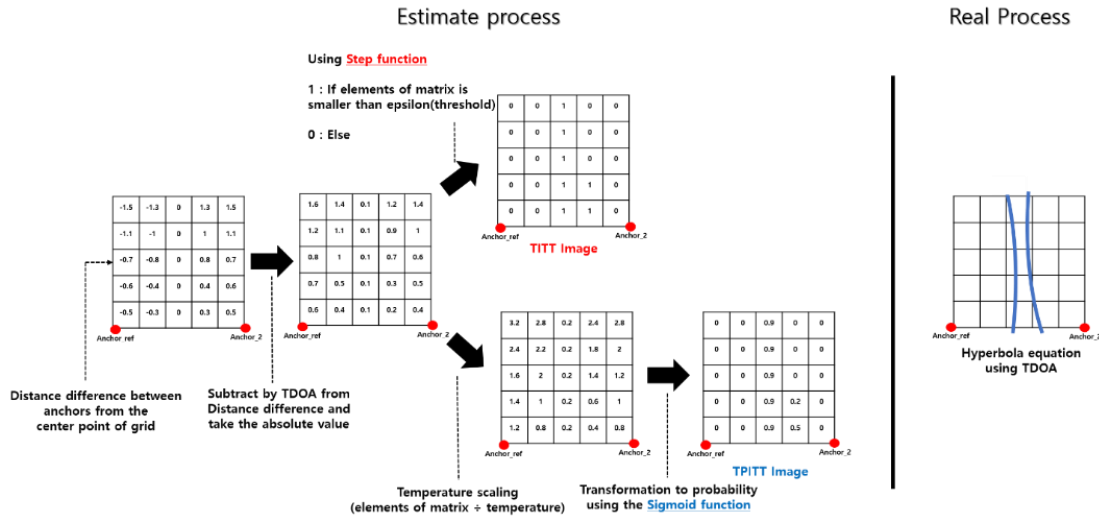


Fig. 2. Conversion of TDoA to image in TITT(upper, red) and TPITT(lower, blue).

$$C_{i,j} = A + [(0.5 + i) \times \text{Grid size}, (0.5 + j) \times \text{Grid size}] \quad (1)$$

이렇게 구한 M1, M2 행렬들을 빼 격자 중점과 앵커 간의 TDoA 행렬 M을 생성한다. 신호를 이미지로 변환하는 과정 2단계는 행렬 M의 각 원소에 실제 TDoA 값을 뺀 뒤, 절대 함수를 적용한다. 즉, 과정 2는 실제 TDoA 값과 중점 좌표를 통해 구한 TDoA 값이 얼마나 유사한 지 비교하기 위해 둘의 차이를 구하는 과정이다. TITT 방식의 TDoA 이미지 전환 방법은 과정 2에서 구한 행렬 M의 각 원소를 최소 거리 차이 ε 과 대소를 비교하여 각 원소를 작거나 같으면 1, 크면 0으로 채워 넣는 구간 함수를 사용한다.

본 논문에서 제안하는 TPITT 이미지 전환 방법은 과정 2를 통해 구한 행렬 M의 원소에 **시그모이드** 함수를 이용해 연속 함수로 바꾸는 과정이다. 이때, temperature 상수 T를 활용해 **스무딩** (smoothing) 정도를 조절한다^[18]. 행렬 M의 중점 좌표를 C, TDoA 값을 s, 기준 앵커의 좌표를 A, 나머지 앵커의 좌표를 A', 최소 거리 차이를 ε , Temperature를 T라고 하면 TPITT TDoA 이미지의 픽셀들은 Equation 2과 같이 나타낼 수 있다.

$$M_{i,j} = 1/1 + e^{\frac{[(C_{i,j}-A)-(C_{i,j}-A')-s]-\varepsilon}{T}} \quad (2)$$

즉, Equation 2를 통해 각 격자에 물체가 존재할 확률을 행렬의 원소에 표현한 TPITT TDOA 이미

지를 구할 수 있다. 이러한 과정은 AWGN을 UWB 신호로부터 구분하기 위해 사전에 학습한 **마스크**와 **컨볼루션** (convolution) 연산을 진행하는 Time-Frequency Masking 방법처럼^[14] 불필요한 정보를 제거하고 특정 공간에서의 타겟의 위치를 세밀하게 표현이 가능하도록 한다.

이와 같은 방식으로 각각의 앵커에서 특정 시간 t에 측정된 TDoA 값을 이미지로 변환하고 합쳐져 3개의 채널을 가진 **tensor**로 **변환**한다. 이때, 각 채널은 물체를 위치를 추적하기 위해 필요한 최소 TDoA 3개로 만들어진 이미지이다.

2.2.3 4D tensor construction with time window

UWB 센서를 이용해 실시간으로 데이터가 앞선 이미지로 변환하는 과정을 마치고 3차원의 배열 (가로, 세로, 채널)로 수집된다. 3차원 형태의 데이터를 Convolutional-LSTM에서 Convolutional filter의 입력 값으로 사용하기 위해 4차원의 시계열 데이터 형태로 변환하는 과정이 필요하다. 변환 결과, 모델의 학습에 사용되는 데이터의 최종적인 형태는 (윈도우, 가로, 세로, 채널)이다.

2.2.4 Target tracking using deep learning

TPITT는 실시간으로 이동 중인 물체의 위치 추정을 위해 Convolutional-LSTM 모델을 활용한다. 해당 모델은 4차원의 시계열 TDoA 이미지 데이터를 입력으로 하고 물체의 x, y 좌표 벡터를 출력 값으로 한다. 모델 학습을 위한 **손실** (loss)

함수로 Mean Square Error (MSE)를 사용한다. 이때, Equation 3에서 N 은 Dataset의 크기, y 는 물체의 실제 공간 좌표, \hat{y} 은 모델이 예측한 물체의 공간 좌표이다.

$$L_{mse} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3)$$

III. 실험

3.1 Environment setting & data generation

제안하는 모델의 성능 검증을 위해 컴퓨터 모의실험을 수행했다. 본 실험은 단일 물체에 대해 위치를 추정하기 위해 전체 공간 (Whole, 기준 센서를 왼쪽 하단 꼭지점으로 크기가 $10m * 10m$ 의 정사각형 공간)과 좁은 공간 (Narrow, 기준 센서를 하단 왼쪽 꼭지점으로 크기가 $2m * 2m$ 의 정사각형 공간)을 실험 공간으로 설정했다. 앵커는 모든 공간의 각 꼭지점에 위치하고 물체의 움직임을 추적하고 있으며, 그 중 하나가 기준 센서로 선택된다고 가정했다. 채널 모델은 IEEE 802.15.4a의 사무실 내 환경을 가정하며^[3] “Bitcraze”사의 “loco positioning system” 모듈을^[19] 이용해 가상 UWB 신호가 만들어진다. 각 센서는 ToA를 기반으로, TDoA는 기준 센서와 나머지 센서 간의 ToA 차이로 중앙 컴퓨팅 유닛 (CCU)에서 계산됐다. 계산 과정에서 TDoA 측정은 millisecond 단위로 이루어지며 각 센서와 CCU 사이의 통신 지연은 없다고 가정했다.

실험을 위한 다양한 환경을 가정하기 위해 각 환경에 따라 각각 다른 측정 오차를 적용했다. 이때, 측정 오차는 환경에 따라 1) AWGN이 극심한 환경 (Noise extreme): 평균 0, 표준 편차가 2, 2) AWGN이 높은 환경 (Noise high): 평균 0, 표준 편차가 1, 3) AWGN이 적은 환경 (Noise low)은 평균 0, 표준 편차가 0.2로 정의되었다.

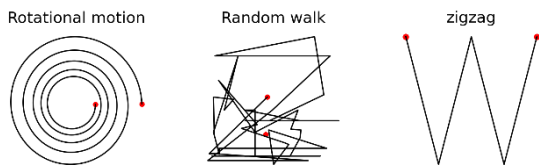


Fig. 3. Example of target's moving.

각 환경에서 물체의 움직임에 따른 TDoA 데

이터를 생성하였다. 이때, 데이터는 물체가 지그재그, 랜덤, 회전 운동을 한다고 가정하며 운동 예시는 Figure 3과 같다. 물체는 가속도 운동을 진행하며 가속도는 표준정규분포를 통해 바뀐다. 이때, 속도의 최대값은 인간의 평균 달리기 속도인 $5m/s$ 으로 가정했다.

물체의 운동 조건이 정해졌다면 학습용 데이터인 랜덤으로 운동하는 물체의 데이터 16,000개를 생성했다. 그런 뒤, 모델 평가용 데이터로 전체 공간에서 지그재그, 랜덤, 회전 운동하는 물체의 데이터 4,000개, 기준 앵커 주변의 좁은 공간에서만 지그재그, 랜덤, 회전 운동하는 물체의 데이터 4,000개를 생성했다. 즉, 모델은 랜덤으로 운동하는 물체의 데이터로만 학습을 진행하며, 전체 공간 혹은 앵커 주변의 좁은 공간에서 지그재그, 랜덤, 회전 운동하는 물체 총 6개의 각각 다른 환경에 대해서 검증 실험을 진행했다.

3.2 Model hyper Parameter

TPITT Convolutional-LSTM 모델을 검증하기 위해 비교군 모델은 총 4개를 설정하였다 (FCNNs, TPITT CNN, TITT CNN, TITT Convolutional LSTM). 이때, Convolutional-LSTM 모델의 구조는 3번의 convolution (Channel & Kernel: $4 \times 11 / 8 \times 8 / 16 \times 4$)과 max pooling (Kernel: 2)를 거친 다음 batch normalization을 한 후, 3번의 dense layer ($128 / 32 / 2$)를 거친 구조이다. CNN 모델의 구조는 Convolutional-LSTM 층이 일반 Convolution 층으로 바뀐 것을 제외하고 모든 구조가 동일했다. FCNNs 모델은 입력값이 TDOA 3개이며 hidden layer는 16, 8 이고 output layer는 2인 구조를 가지고 있다. 모델의 하이퍼파라미터는 Table 1과 같이 5개 모델에 공통적으로 적용했다.

Table 1. The hyper parameters of models

| Parameter | Value |
|---------------------|---------------|
| Activation function | Relu |
| Optimizer | Adam |
| Learning rate | $1 * 10^{-3}$ |
| Loss function | MSE |
| Drop out | 0.6 |
| Mini batch size | 20 |
| Epoch | 30 |

Table 2. Average of the Euclidean distances between predictions and actual coordinates due to environmental changes

| Environment | Model | Random Whole | Random Narrow | Zigzag Whole | Zigzag Narrow | Rotation Whole | Rotation Narrow |
|---------------|-----------------|---------------|---------------|---------------|---------------|----------------|-----------------|
| Noise low | FCNNs | 0.4844 | 1.0091 | 0.6231 | 1.0292 | 0.4029 | 0.8156 |
| | CNN-TITT | 0.5624 | 0.6003 | 0.8476 | 0.613 | 0.3774 | 0.5476 |
| | CNN-TPITT | 0.5389 | 0.5843 | 0.818 | 0.7273 | 0.3775 | 0.547 |
| | Conv_LSTM-TITT | 0.2376 | 0.3378 | 0.3296 | 0.3549 | 0.1975 | 0.295 |
| | Conv_LSTM-TPITT | 0.2356 | 0.3309 | 0.3377 | 0.3534 | 0.1883 | 0.2924 |
| Noise high | FCNNs | 0.7929 | 1.0627 | 0.9081 | 1.1912 | 0.7883 | 1.0163 |
| | CNN-TITT | 1.0224 | 0.8345 | 1.3036 | 0.9705 | 0.9385 | 0.7645 |
| | CNN-TPITT | 0.958 | 0.7334 | 1.2493 | 0.9651 | 0.8694 | 0.689 |
| | Conv_LSTM-TITT | 0.7256 | 0.6336 | 0.8908 | 0.6995 | 0.6128 | 0.5429 |
| | Conv_LSTM-TPITT | 0.6932 | 0.5524 | 0.8266 | 0.6122 | 0.5661 | 0.4732 |
| Noise Extreme | FCNNs | 1.3184 | 1.3186 | 1.4579 | 1.2931 | 1.3531 | 1.2488 |
| | CNN-TITT | 1.6233 | 1.1766 | 1.968 | 1.1185 | 1.5025 | 1.0349 |
| | CNN-TPITT | 1.5351 | 0.9876 | 1.8395 | 1.0656 | 1.4151 | 0.9072 |
| | Conv_LSTM-TITT | 1.2309 | 0.9234 | 1.4571 | 0.9863 | 1.1168 | 0.7872 |
| | Conv_LSTM-TPITT | 1.1435 | 0.7647 | 1.2356 | 0.8702 | 1.0039 | 0.6326 |

Table 3. Standard deviation of the Euclidean distances between predictions and actual coordinates due to environmental changes

| Environment | Model | Random Whole | Random Narrow | Zigzag Whole | Zigzag Narrow | Rotation Whole | Rotation Narrow |
|---------------|-----------------|---------------|---------------|---------------|---------------|----------------|-----------------|
| Noise low | FCNNs | 0.1153 | 0.3154 | 0.1773 | 0.4123 | 0.2554 | 0.3712 |
| | CNN-TITT | 0.0678 | 0.0706 | 0.0926 | 0.0495 | 0.0702 | 0.095 |
| | CNN-TPITT | 0.0616 | 0.1085 | 0.1683 | 0.1946 | 0.0664 | 0.1916 |
| | Conv_LSTM-TITT | 0.0104 | 0.0435 | 0.0348 | 0.0424 | 0.021 | 0.0612 |
| | Conv_LSTM-TPITT | 0.0179 | 0.0931 | 0.0573 | 0.0932 | 0.0242 | 0.0741 |
| Noise high | FCNNs | 0.0249 | 0.3147 | 0.0444 | 0.2585 | 0.0828 | 0.3256 |
| | CNN-TITT | 0.0661 | 0.0735 | 0.0956 | 0.0719 | 0.1081 | 0.0816 |
| | CNN-TPITT | 0.0452 | 0.054 | 0.1123 | 0.0809 | 0.0911 | 0.0798 |
| | Conv_LSTM-TITT | 0.0173 | 0.0447 | 0.0513 | 0.0793 | 0.0282 | 0.084 |
| | Conv_LSTM-TPITT | 0.0226 | 0.0176 | 0.0355 | 0.0687 | 0.0402 | 0.0257 |
| Noise Extreme | FCNNs | 0.012 | 0.1334 | 0.0916 | 0.1283 | 0.0477 | 0.161 |
| | CNN-TITT | 0.0493 | 1.2644 | 0.1249 | 0.1048 | 0.1021 | 0.1487 |
| | CNN-TPITT | 0.0582 | 0.0824 | 0.225 | 0.0846 | 0.0478 | 0.0851 |
| | Conv_LSTM-TITT | 0.0545 | 0.0671 | 0.0713 | 0.0586 | 0.0414 | 0.1079 |
| | Conv_LSTM-TPITT | 0.056 | 0.0413 | 0.2998 | 0.0376 | 0.0599 | 0.0188 |

3.3 Model comparison

검증 실험의 평가지표로는 예측과 실제 사이의 유클리디안 거리를 사용하였고, 10번의 반복 실험을 진행하였다. 실험 결과는 Table 2와 3과 같다. Table 2를 살펴보면 모든 실험에서 Convolutional-LSTM 모델이 모든 상황에서 평균 0.2 ~ 0.5m 좋은 성능을 보였으며 이는 모든 모델 중 가장 좋은 예측을 하였다. 분산 또한 Table 3에서 확인할 수 있는데, Noise Extreme & Random Whole 상황을 제외한 모든 상황에서 Convolutional-LSTM이 다른 모델에 비해 좋은 결과를 보였으며, 최대 0.05만큼 낮은 분산 값을 보였다. Convolutional-LSTM 모델은 시계열 정보를 활용함으로써 물체가 학습 시와 다른 움직임을 보이는 경우에도 위치를 잘 파악할 수 있었다.

Figure 4는 Noise Extreme 상황에서 Convolutional-LSTM과 FCNNs 모델이 예측한 결과를 좌표 평면에 시각화 한 것이다. Figure 4의 좌측에 존재하는 그래프들을 살펴보면 두 모델

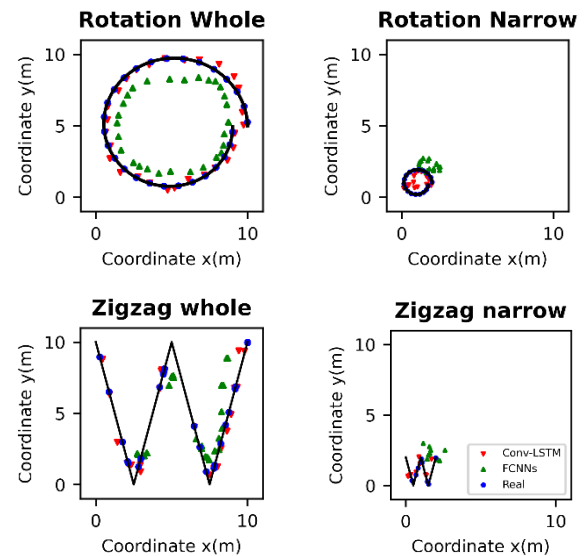


Fig. 4. Example of target tracking rotating and zigzag objects using FCNNs and TPITT

의 예측 값과 실제 물체의 위치와 큰 차이를 보이지 않았다. 하지만 앵커 주변 (narrow)으로 갈수록 물체의 운동과 관련 없이 큰 오차를 보이고 있으며 이는 Table 2와 3를 통해 더욱 자세히

알 수 있다. FCNNs 모델은 좁은 공간에서 진행한 모든 상황에서 가장 좋지 않은 결과를 보인다. 또한, FCNNs 모델은 오차가 거의 없는 공간에 대해서도 평균 오차가 1m가 넘는 결과를 보이는데 좁은 공간이 가로 세로 2m의 좁은 공간임을 고려했을 때 매우 좋지 않는 성능을 보였다. 이를 통해 이미지 기반의 모델은 숫자 기반의 모델과 다르게 보편적인 상황만 학습을 진행하여도 다양한 상황 또한 잘 예측한다는 사실을 알 수 있었다. 즉, 이미지 기반의 모델은 학습하지 않은 상황에서도 강건하고 섬세하게 예측을 하였다.

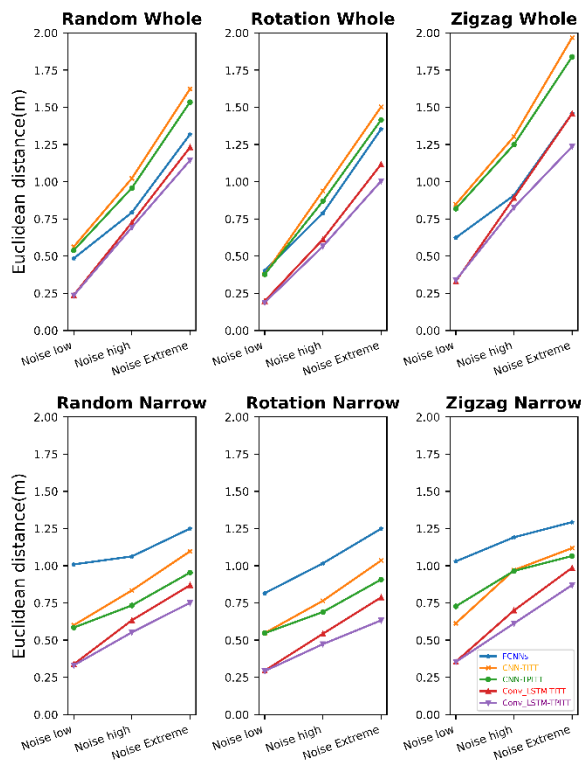


Fig. 5. Results of Euclidean distance between prediction and actual coordinates according to the magnitude of AWGN

Figure 5는 AWGN의 변화에 따른 모델들의 평균 변화 그래프이다. 그래프를 자세히 살펴보면 AWGN이 거의 없는 공간에서는 TITT와 TPITT는 거의 비슷한 성능을 보였다. 하지만 AWGN이 많은 공간인 Noise high 상황에서의 평균 차이는 0.03 ~ 0.08m이며 AWGN이 극심한 공간인 Noise extreme 상황에서의 평균 차이는 0.09 ~ 0.22m이다. 즉, AWGN이 증가하는 상황일수록 TITT와 TPITT의 성능 차이가 더욱 커지는 경향을 보인다.

다. 이를 통해, TPITT의 확률화 과정은 이미지 변환 방식의 주요 목적인 AWGN을 제거하는 역할을 더 잘 수행한다는 사실을 알 수 있다.

IV. 결 론

본 연구는 이미지 변환 과정을 통해 AWGN을 감소시키고, 다양한 타겟과 환경에서 일반적으로 사용할 수 있는 위치 추적 방법인 TPITT 모델을 제안하였다. 실험 결과 TDOA 이미지 기반의 TPITT 모델은 FCNNs 모델과 비교했을 때, 일반적이지 않은 다양한 상황에서 안정된 성능을 보였다. 또한, AWGN이 많은 환경에서 TITT보다 더 낮은 예측 오류를 보였다. 이는 TPITT의 확률화 과정이 AWGN에 대한 강건성을 향상시킬 수 있음을 의미한다. 마지막으로, 다른 모델들과 비교하여 Convolutional-LSTM 모델이 전반적으로 가장 우수한 성능을 보였다. UWB 신호의 시계열 정보를 활용하는 것이 효과적임을 알 수 있었다.

본 연구는 딥러닝 모델을 기반으로 하고 있으므로, 다음의 한계점을 가지고 있다. 학습에 사용될 수 있는 TDoA 데이터베이스가 존재하지 않거나 그 양이 매우 적은 경우 높은 성능을 얻기 어렵다. 추후 연구로서, 다른 대량의 데이터로 이미 학습된 모델을 활용하는 방식인 전이 학습 (Transfer Learning)이나, 다른 환경에서 학습된 모델을 새로운 환경에 맞게 조정하는 도메인 적응 기법 (Domain Adaptation)을 도입해보고자 한다. 이를 통해 TDoA 데이터베이스 구축 과정을 간소화한다면 제안 방법의 활용성이 증대될 것이다.

References

- [1] Wells, D., Beck, N., Kleusberg, A., Krakiwsky, E. J., Lachapelle, G., Langley, R. B., ... & Delikaraoglou, D. (1987). Guide to GPS positioning. In Canadian GPS Assoc.
- [2] Rizos, C. (2013, March). Locata: A positioning system for indoor and outdoor applications where GNSS does not work. In Proceedings of the 18th

- Association of Public Authority Surveyors Conference (pp. 73-83).
- [3] 장병준. (2022). UWB 측위기술의 원리 및 동향. 한국전자과학회논문지, 33(1), 1-11.
- [4] Fontana, R. J., & Richley, E. A. (2007, September). Observations on low data rate, short pulse UWB systems. In 2007 IEEE International Conference on Ultra-Wideband (pp. 334-338). IEEE.
- [5] Zafari, F., Gkelias, A., & Leung, K. K. (2019). A survey of indoor localization systems and technologies. IEEE Communications Surveys & Tutorials, 21(3), 2568-2599.
- [6] Yan, B., & Xiaochun, L. (2009, August). Research on UWB indoor positioning based on TDOA technique. In 2009 9th International Conference on Electronic Measurement & Instruments (pp. 1-167). IEEE.
- [7] er Andersen, A. C. (2011). Comparative analysis of multilateration methods for signal emitter positioning.
- [8] Krishnan, S., Santos, R. X. M., Yap, E. R., & Zin, M. T. (2018, November). Improving UWB based indoor positioning in industrial environments through machine learning. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 1484-1488). IEEE.
- [9] Wang, Z. Q., Zhang, X., & Wang, D. (2018, August). Robust TDOA Estimation Based on Time-Frequency Masking and Deep Neural Networks. In Interspeech (pp. 322-326).
- [10] Poulose, A., & Han, D. S. (2020). UWB indoor localization using deep learning LSTM networks. Applied Sciences, 10(18), 6290.
- [11] Wang, H., Wang, X., Xue, Y., & Jiang, Y. (2020, June). UWB-based Indoor Localization Using a Hybrid WKNN-LSTM Algorithm. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (Vol. 1, pp. 1720-1725). IEEE.
- [12] Niitsoo, A., Edelhäuser, T., & Mutschler, C. (2018, September). Convolutional neural networks for position estimation in tdoa-based locating systems. In 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN) (pp. 1-8). IEEE.
- [13] Huang, L., Li, J., Hao, H., & Li, X. (2018). Micro-seismic event detection and location in underground mines by using Convolutional Neural Networks (CNN) and deep learning. Tunnelling and Underground Space Technology, 81, 265-276.
- [14] Yilmaz, O., & Rickard, S. (2004). Blind separation of speech mixtures via time-frequency masking. IEEE Transactions on signal processing, 52(7), 1830-1847.
- [15] J. Baek, J. Lee, H. Shim, S. Im and Y. Han, "Target tracking initiation for multi-static multi-frequency PCL system", IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 10 558-10 568, 2020.
- [16] Baek, J., Lee, S., Lee, C., & Park, S. (2021, October). A Deep Learning Approach for Robust Target Tracking in a Cluttered Environment. In 2021 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 1033-1035). IEEE.
- [17] Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." Advances in Neural Information Processing Systems. 2015
- [18] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, July). On calibration of modern neural networks. In International conference on machine learning (pp. 1321-1330). PMLR.
- [19] Alexis PAQUES, MultilaterationTDOA (2019), GitHub repository, <https://github.com/AlexisTM/MultilaterationTDOA>
- [20] Szeliski, R., & Shum, H. Y. (1997, August). Creating full view panoramic image mosaics and environment maps. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques (pp. 251-258).