A Hybrid Approach to Impulse-Radio UWB Self-Localization with Passive Reflectors

Kotchapansompote, Palakon; Kumchaiseemak, Nakorn; Kaemarungsi, Kamol; Suwajanakorn, Supasorn

DOI
10.1109/JSEN.2024.3449376

Publication date
2024

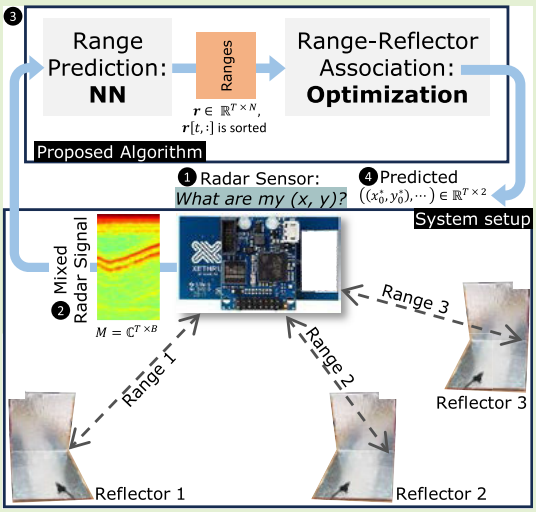
Document Version
Final published version

Published in
IEEE Sensors Journal

A Hybrid Approach to Impulse-Radio UWB Self-Localization With Passive Reflectors

Palakon Kotchapansompote, Nakorn Kumchaiseemak, *Member, IEEE*, Kamol Kaemarungsi, *Senior Member, IEEE*, and Supasorn Suwajanakorn

***Abstract*—This study introduces a simple and cost-effective localization system using a moving ultrawideband (UWB) radar sensor and passive reflectors at fixed points. Our use of UWB radar ensures consistent performance across various lighting conditions and offers privacy protection. Our hybrid pipeline first predicts the ranges from the sensor to the reflectors and then predicts the radar's position from these ranges. Two key components of the hybrid pipeline are a neural network for range prediction and an optimization-based "association" step. The neural network solves the challenge of predicting individual ranges from the mixed radar signal, while the association step matches each predicted range with its corresponding reflector using a novel regularized trilateration formulation. Experiments validated our approach, yielding an average positional estimation error of approximately 0.20 m, making it suitable for human or robot tracking applications. Our contributions include a novel localization setup, an algorithm, and a real-world UWB dataset with annotated 3-D ground-truth positions.**



***Index Terms*— Localization, neural network, radar, Transformer, ultrawideband (UWB), UNet.**

I. Introduction

SELF-LOCALIZATION is an essential first step for a robot or an autonomous agent to effectively interact with its environment. Various sensing techniques were used for this purpose, such as visual sensors, ultrasonic sensors (US), light sensors, and radar sensors. In this work, we explore the use of ultrawideband (UWB) radar for self-localization in indoor settings. Key advantages of radar sensors include their robustness in adverse lighting conditions compared with visual- or light-based sensors [1], [2] and their enhanced privacy protection, as they do not capture easily identifiable information, such as facial images. In addition, short-range radar sensors offer benefits, such as lower power consumption, reduced silicon area, simpler design and control, and decreased data processing [3].

Previous radar localization studies have utilized various setups from sophisticated and costly hardware customizations to advanced multitransceiver (MIMO) radar systems. For instance, studies in [4], [5], and [6] have demonstrated human tracking capabilities through the use of custom-designed radio antennas, circuitry, amplification, and filtering. The study of [1] also integrated an inertial measurement unit (IMU) to enhance indoor self-localization accuracy. With MIMO radar, the estimation of self-velocity can be addressed by analyzing Doppler shifts from the environment using Hough transform [7] or by separating translational and rotational movements [8], and self-localization can be achieved using a simultaneous localization and mapping (SLAM)-based technique [9].

In contrast, our system utilizes a much simpler setup of a commercial single-input, single-output (SISO), UWB radar sensor, and three passive reflectors, as shown in Fig. 1. This eliminates the need for custom or MIMO radar systems. We explore the use of passive corner reflectors, which can be

Manuscript received 10 May 2024; revised 24 July 2024; accepted 13 August 2024. Date of publication 30 August 2024; date of current version 2 July 2025. This work was supported by the VISTEC-NSTDA under Grant 53/2563. The associate editor coordinating the review of this article and approving it for publication was Prof. Guido Dolmans. *(Corresponding author: Supasorn Suwajanakorn.)*

Palakon Kotchapansompote and Supasorn Suwajanakorn are with the School of Information Science and Technology (IST), Vidyasirimedhi Institute of Science and Technology (VISTEC), Rayong 21210, Thailand (e-mail: palakon.k_s20@vistec.ac.th; supasorn.s@vistec.ac.th).

Nakorn Kumchaiseemak is with the School of Information Science and Technology (IST), Vidyasirimedhi Institute of Science and Technology (VISTEC), Rayong 21210, Thailand, and also with the MS3 Group, Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: nakorn.k_s18@vistec.ac.th).

Kamol Kaemarungsi is with the National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA), Pathum Thani 12120, Thailand (e-mail: kamol.kaemarungsi@nectec.or.th).

Digital Object Identifier 10.1109/JSEN.2024.3449376

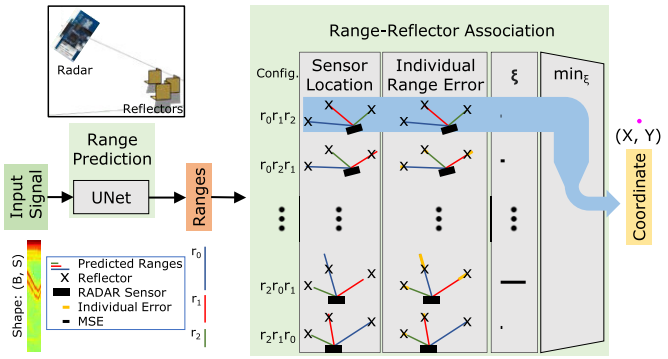


Fig. 1. Proposed radar self-localization system consists of three main steps: the preprocessing step, the range prediction step, and the association step. ξ is the optimization loss.

made from household items, such as aluminum foil and plastic boards, making it easy and economical to add more reflectors for improved accuracy. The variant of UWB radar that we used is the impulse-response (IR) UWB radar, which allows simplified range calculation via time-of-arrival (ToA) and minimizes processing and signal storage [10]. Unlike existing work that explores similar setups solely in simulation [11], [12], our work is the first real-world system under this setup.

Our localization processing pipeline takes the radar signal as input and outputs the position of the radar sensor. It is based on trilateration method; however, our setup presents unique challenges. First, since the radar receives mixed, superposed signals from the three homogeneous reflectors, we must separate the signals to determine individual reflector ranges. Second, we must associate each predicted range with its corresponding reflector for trilateration. To address these challenges, we propose a hybrid pipeline that combines a neural network for range prediction and optimization for regularized trilateration. This pipeline consists of three components.

1) *Preprocessing:* This step calculates the input feature for our neural network by computing the complex modulus of the signal and stacking features from previous timesteps, allowing the network to access historical data.
2) *Range Prediction:* This step predicts N range values from the superimposed radar signal using a neural network. However, these range values are not yet associated with the reflectors.
3) *Association:* This step matches each range value with its corresponding reflector by evaluating all pairings using optimization. Given the optimal pairing, regularized trilateration is performed to estimate the radar's position.

In summary, the contributions of this study are as follows.

1) *Novel UWB Localization Setup:* Our setup features a compact impulse-radio UWB sensor and passive reflectors, offering simple setup, scalability, and low manufacturing costs.
2) *Hybrid Algorithm:* Our hybrid pipeline isolates the task of discerning superimposed signals, allowing the neural network to focus on range extraction, while the trilateration-based optimization efficiently addresses the association challenge.
3) *Position-Annotated UWB Dataset:* The first annotated dataset of UWB radar signals from 240 real *sessions* across eight different indoor scenes with 3-D ground-truth positions. This dataset will be released publicly.

II. Related Works

A. Approaches for UWB Radar Localization

1) Radar Setups:

a) Fixed radar approach: In this approach, multiple radar sensors are installed at fixed positions in the environment. The setup allows for precapturing of the background signal, which can then be subtracted from the input signal during localization. This precapturing process greatly helps simplify the signal-processing pipeline. Classical techniques described in [13] used two radar systems with the *joining intersections of the ellipses* approach to localize a moving object through a wall. Li et al. [14] proposed detecting human NLOS in a cornered corridor with a specific dimension/shape with FMCW radar. The range, azimuth of the range, and the shape of the corner allow for the localization of the human. The work in [15] uses two reconfigurable intelligent surfaces (RISs) to help a radar take advantage of the side lobes. With additional reflections from RISs, more information is presented to the localization pipeline and increases prediction accuracy. Other works, such as [11], [16], used neural networks to predict ranges from sensors to the robot using information, such as received signal strength (RSS), ToA, and AoA, and then used convolutional neural networks (CNNs) [11] or long short-term memory networks (LSTMs) [16] for localization.

b) Moving radar approach: This setup has the radar sensor(s) installed on the robot itself. RadarSLAM [17] adapted an existing acoustic SLAM system (BatSLAM) [18] to work with UWB radar signal. In [19], fixed stationary transmitters emit signals that a robot, aided by IMUs, used to determine their positions via the ToAs. Our work aligns with this moving radar approach and targets scenarios in which the sensor is attached to a human or a robot moving across the ground.

2) Localization Markers:

Position markers for localization can be classified into three categories: 1) environmental features; 2) active markers; and 3) passive markers.

a) Environmental features: This category uses no specific reflectors and relies solely on environmental reflections. It eliminates the need for marker installation, increases flexibility, and remains unobtrusive. However, it requires significant computational resources and extensive training data. An example of such a method is a SLAM-based algorithm presented in [17]. This algorithm establishes correspondences between observed 3-D feature points and subsequent captures to reconstruct the map of the scene.

b) Active markers: Active markers are powered electronic devices capable of amplifying or manipulating radio signals to uniquely identify themselves. While this enhances the signal for ease of detection and facilitates further localization processing, their dependency on power sources can impact maintenance and long-term sustainability [20], [21], [22].

c) Passive markers: These markers do not require any external power source, making them cost-effective and easily scalable. This category is suitable for long-term deployment

without the need for frequent maintenance [23], [24]. While using environmental features has its advantages, there is no guarantee that the environment will always contain distinct features for localization. Therefore, this work opts for the passive reflector scheme but ensures that the reflector design is not overly complicated.

B. Algorithms for UWB Radar Localization

1) Lateration-Based: This class of techniques leverages ToA and angle-of-arrival (AoA) data to estimate distances and angles for localization purposes. In [25], ToA and AoA were jointly estimated in a line-of-sight (LOS) setting. For non-line-of-sight (NLoS) settings, Yu et al. [26] used the Taylor series least-squares method to reduce estimation errors. In addition, an IR-UWB localization system was demonstrated in [27], employing time-difference-of-arrival (TDoA) estimation coupled with an error correction algorithm.

2) Classical Machine Learning-Based: Rana et al. [28] reformulated the localization task as a classification problem using a multiclass support vector machine (SVM). The position of the object was estimated from the class of patterns.

3) Neural Network-Based Classification: In [29], CNNs were proposed for localization, utilizing Wi-Fi's RSS and channel state information (CSI). Khatab et al. [30] performed location fingerprinting, whereas Sinha and Hwang [31] proposed a CNN-based system, both based on RSS. Classification-based systems inherently have discretization errors, especially in large areas with limited discretization bins. A solution is to increase the number of bins, but this approach introduces additional parameters and further complicates the system.

Our method also relies on discrete range bins; however, we utilize these bins to store the probability distribution of the radar's position, computed from the SoftMax of our network's output (see Section IV-B2). This enables us to compute a continuous *expectation* of the position, thereby circumventing the discretization errors associated with naive range-bin classification.

4) Neural Network-Based Regression: Techniques in this category include DNN-based systems [32], deep LSTM schemes [33], and recurrent neural networks (RNNs) [34] that exploit RSS and trajectory data and CNN-based methods [35], [36]. Poulose and Han [16] first obtain ToAs and use an LSTM network for localization. For this work, we propose a hybrid approach that combines neural networks with an optimization-based algorithm. This combination allows the neural network to focus on estimating ranges, while the optimization tackles only the conversion from ranges to positions based on physical trilateration constraints.

C. Datasets of UWB Radar Localization

In the literature, multiple datasets exist for UWB localization research. The dataset from [37] focused on the detection of human victims behind obstacles, such as walls and doors. Maitre et al. [38] offered a dataset containing 15 activities of daily living (ADLs) captured from participants in real-world settings. Ahmed et al. [39] compiled data on 12 dynamic hand gestures from nine volunteers. Zhengliang et al. [40] provided a dataset for training neural networks to detect human motion status through walls. Our work also provides a dataset of real-world UWB radar signals with 3-D ground-truth positions. To the best of our knowledge, this dataset is the first of its kind.

(a) (b) (c)

Fig. 2. Capture setup. (a) System was set up for dataset collection with three radar corner reflectors and a radar sensor attached to the VR headset. A laptop was used to record the radar signal through a USB cable. (b) Radar sensor is mounted on the Oculus Quest 2 for ground-truth position. (c) Average reflector pattern is depicted with reflectors marked by blue crosses, the main axis by a red dashed line, and the central line in solid green.

TABLE I
SETTINGS FOR IR-UWB: NOVELDA XETHRU X4M03

Setting	Value
Range Resolution	0.05144 m
Sampling Frequency	140 Hz
Operating Frequency	$6-8$ GHz
Maximum Range	9.00 m
Minimum Range	0.00 m
Field of View	73°

III. SYSTEM SETUP

This work addresses the self-localization of an IR-UWB radar sensor [41] within a scene containing three passive radar reflectors ($N = 3$) placed at known positions. We assume that the radar sensor only experiences horizontal 2-D motion. This is relevant to scenarios where we attach the sensor to a ground robot or a human that moves across the floor at a fixed height.

A. Hardware Setup

To localize our radar sensor in a 2-D space, we place $N = 3$ passive reflectors in the scene. However, our system is based on neural networks, which are trained using ground-truth positions. To capture these ground-truth positions and create our training set, we designed a position capture system based on a virtual reality (VR) headset, which provides full three-dimensional 3-D positions of our radar sensor, as shown in Fig. 2. The captured scenes are depicted in Fig. 3. The details of the hardware used in our study are as follows.

1) *Radar Sensor:* A commercial off-the-shelf (COTS), single-transceiver IR-UWB radar sensor is based on Novelda XeThru X4M03 [41]. The settings of this radar sensor are summarized in Table I. The sensor was positioned with the reflectors in its view and captured the reflected radar signals and the environmental noises.
2) *Reflectors:* We fabricated several identical radar corner reflectors, which reflect signals directly back toward their source. These reflectors were made from plastic boards with a side width of 0.35 m, lined with aluminum foil to enhance signal reflection [42].

We arranged the three reflectors in a triangular shape, with average side lengths of ±1.72, ±1.27, and ±0.58 m from the longest to the shortest sides, respectively. The average angle between the two shorter sides was 134.50°. These patterns do not need to be identical, which helps simulate realistic installation scenarios. The average positions of the reflectors, along with their general areas, are visualized in Fig. 2(c).

B. Radar Signal Characteristics

The setting for radar data acquisition is shown in Table I. In each timestep, the IR-UWB radar sensor recorded reflected radar signals as complex values for $B = 177$ *range bins*. Each bin is equally spaced from the *minimum* to the *maximum* ranges. Timestamps were also captured. The resulting radar frame is first preprocessed as described in Section IV-B1. Then, it is used to predict the final radar position.

IV. Methodology

Our localization system relies on trilateration. This technique requires multiple known reference points and distances from the target to each reference point. Based on this information, the trilateration technique estimates the target's position to match the inputs. In this work, the known reference points are the positions of the radar reflectors. Distances, or ranges, to the reflectors were estimated using a neural network, which processes the radar frame, $\mathbf{M}$, and outputs three ranges. However, it does not specify which range is associated with which reflector. This leads to a problem that we call *association ambiguity*. To address this, we developed a localization pipeline that resolves trilateration from all possible permutations of reflectors and range values.

A. Data Collection

The dataset in this work consists of 240 sessions from eight scenes, as shown in Fig. 3. Each timestep in a session includes a timestamp, the radar frame, and the radar's position. In Sections IV-A1 and IV-A2, we will discuss the methods used to collect the data and some statistics about the dataset.

1) Radar and Position Data: As shown in Fig. 2(b), we attached our UWB radar sensor to the top of an Oculus Quest 2 VR headset and simultaneously captured both the radar signal and the headset's position, which was tracked automatically by the headset. We then computed the radar's position by adding a fixed vertical offset to the headset's position according to the physical displacement. As the tracking and radar signal were captured at different frame rates, we time-synchronized them by upsampling the 90-ft/s tracking sequence to match the 140-ft/s radar signal.

2) Dataset Statistics: Our dataset consists of 240 real capturing sessions across eight indoor scenes. Each scene contains 30 sessions, each lasting approximately 60 s. These sessions were captured by a person holding the headset with the mounted UWB radar at eye level and continuously walking around at a slow pace. A total of ≈14 400 s of data have been acquired. The system's *range resolution* (the width of a range bin) is approximately 0.05 m. Table II shows basic information about the captured scenes.

Fig. 3. Scenes 1–8 of the collected data.

TABLE II
Characteristics, Data Points, and Radar Range of Each Scene

Scene No.	#Data Points	Radar Range
1	247,981	1.5 – 5.6 m
2	247,984	1.6 – 5.8 m
3	247,982	1.7 – 6.60 m
4	247,968	1.5 – 6.50 m
5	247,727	1.6 – 5.7 m
6	247,612	1.5 – 6.00 m
7	247,977	1.5 – 7.20 m
8	239,641	1.6 – 7.70 m

B. Pipeline

Our hybrid pipeline consists of three components: preprocessing, range prediction, and association step.

1) Preprocessing: The raw input radar frame, $M \in \mathbb{C}^{T\times B}$, is a complex-valued array that covers T timesteps and B range bins. We first convert M into a real array of its complex modulus, $M' \in \mathbb{R}^{T\times B}$, where $M'_{(t,b)} = |M_{(t,b)}|$. To support real-time applications, our network processes each time step independently without requiring access to the entire array of T timesteps. To provide additional temporal information for the network, we expand M' to a 3-D tensor $M'' \in \mathbb{R}^{T\times B\times S}$, where S is the size of the temporal window and the last dimension of M'' contains the data of the previous S timesteps, i.e.,

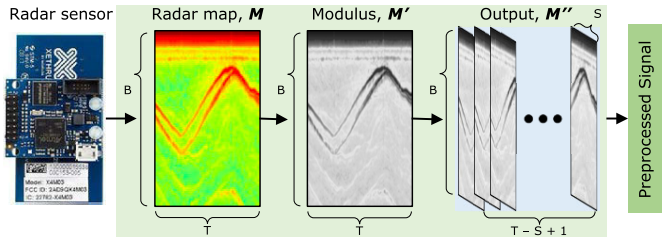


Fig. 4. Preprocessing step computed the modulus of the complex radar frame M to obtain M'. This M' was then expanded into a 3-D tensor, M''. M'' is also called the preprocessed signal, $\mathbf{I}$.

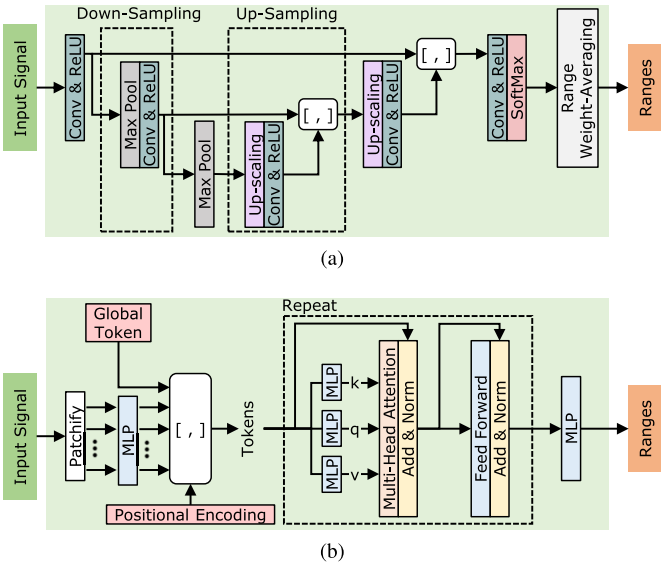


Fig. 5. Proposed neural network for range estimation: (a) UNet and the BL and (b) Transformer network architectures.

$M''_{(t,b,i)} = M'_{(t-S+i,b)}$ (1-based indexing). At each timestep t, the network processes $M''_t \in \mathbb{R}^{B\times S}$. Fig. 4 illustrates this step.

2) Range Prediction Using Neural Network: In this step, we predict the ranges from the radar to N reflectors, denoted by $\hat{\mathbf{r}}(t) \in \mathbb{R}^N$ from our preprocessed signal M''. To achieve this, we propose using a neural network based on the UNet architecture [43], which is trained on our own dataset. In our experiment, we compare this architecture choice with the Transformer [44], another popular architecture.

a) UNet model: Our UNet model, depicted in Fig. 5(a), consists of the following components in sequence: initial convolution, downsampling, upsampling, and final prediction. The unique approach of the UNet architecture involves skip connections that link the downsampling and upsampling layers directly, facilitating feature reuse from earlier layers. For each timestep t, the input $M''_t \in \mathbb{R}^{B\times S}$ is treated as a 1-D input of length B with the number of channels equals S. Below are brief descriptions of each component in the UNet model.

1) *Initial Convolution:* This component involves two consecutive groups of 1-D convolution operation, batch normalization (BN), and rectified linear unit (ReLU) activation. *BN* [45] normalizes inputs to ensure consistent distributions. This process accelerates the training process and reduces sensitivity to network initialization. *ReLU*, a common *activation function* known for its simplicity, introduces nonlinearity into neural networks, facilitating the learning of complex patterns. We abbreviate these groups as Conv-BN-ReLU. The convolution is performed using a 1-D kernel of size three along the range axis. This operation increases the number of channels from $S \rightarrow 256$ after the first Conv-BN-ReLU, and then, it remains at 256 channels after the second Conv-BN-ReLU.
2) *Downsampling:* This component applies an operation group (MaxPool, 2 × Conv-BN-ReLU) n times. *MaxPool* is a downsampling strategy, which preserves critical features by retaining the maximum value in each sliding window, whose size is given by the *window size*. The window is moved by the *stride* parameter. It enhances the network's robustness to input variations and decrease computational complexity. Following the original UNet architecture structure, the 1-D MaxPool operation halves the signal dimension along the range axis using a window size of two and a stride of two. The result is then fed through two sets of Conv-BN-ReLU, which double the number of channels from ch $\rightarrow$ 2ch $\rightarrow$ 2ch, where ch is the input channel to each group. This process allows the information to regroup, without significantly reducing the number of parameters.
3) *Upsampling:* The upsampling component is similar to the downsampling component, but the operation is in reverse. This component uses 1-D *transposed convolutions* (TransConv) [46] to double the number of range bins and repeats the (TransConv, 2 × Conv-BN-ReLU) operation group n times. This operation decreases the number of channels from ch $\rightarrow$ (ch/2) $\rightarrow$ (ch/2). The outputs from the previous upsampling group and the respective downsampling group will have the same shape. Then, they are concatenated and fed into the next upsampling group. This upsampling process is also repeated n times.
4) *Final Prediction:* This component consists of a convolutional layer that outputs a $B \times N$ array, where B and N represent the number of range bins and reflectors, respectively. Then, SoftMax is applied to the output along the B-axis, which computes the probability of each reflector being located in each of the B range bins. The dimension remains $B \times N$, with each column vector now summing up to one. We convert these probability values into actual range values by computing the expectation. Specifically, the range for reflector i is calculated as $\hat{\mathbf{r}}_i = \sum_{b=1}^{B} R_b \times p_{b,i}$, where R_b is the known range of bin b and $p_{b,i}$ is the output probability from SoftMax.

b) Transformer model: Our Transformer model is illustrated in Fig. 5(b). It is a scaled-down version of the encoder part of the Transformer architecture in [44]. It divides the M'' (preprocessed signal) along the range axis into smaller patches or *tokens*. These tokens are individually processed through a two-hidden-layer multilayer perceptron (MLP). Following [44], an additional *global token* is included, and standard positional encodings are appended in the channel dimension. Our model comprises repeated encoder blocks, where each block has multiple attention heads and a feedforward layer.

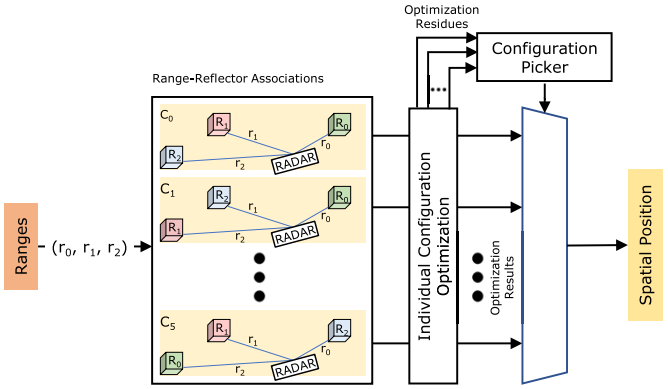


Fig. 6. Association step: given reflector positions (R_0, R_1, R_2) and the predicted ranges (r_0, r_1, r_2), this step determines the correct *configuration* by examining their optimization error and outputs the predicted spatial position.

At the final layer, only the global token is processed through a two-hidden-layer MLP to yield N output ranges.

We trained both UNet and Transformer models with the same supervised loss function

$$L = \mathbb{E}_t\left[\|\hat{\mathbf{r}}(t) - \mathbf{r}(t)\|_2^2\right] \tag{1}$$

where $\hat{\mathbf{r}}(t) \in \mathbb{R}^N$ consists of the predicted ranges and $\mathbf{r}(t) \in \mathbb{R}^N$ consists of the ground-truth ranges. The elements in $\mathbf{r}(t)$ are sorted in ascending order.

3) Reflector-Range Association: This step aims to translate the predicted ranges into the radar position. However, this task is nontrivial in our setup, because there is no direct association between each predicted range and its corresponding reflector. We refer to this challenge as the *association ambiguity* problem. To solve this, we examine all possible $N!$ association configurations of N reflectors in each time step t. Then, for each config, we solve a *regularized* trilateration problem with a novel *association* loss function. The config with the lowest loss value will be used to produce the final position prediction. Fig. 6 illustrates the overview.

To formally explain the algorithm, let an association config $c \in \mathbf{C} = \mathrm{Sym}(\{1, 2, \ldots, N\})$ be a permutation function that maps the ith reflector to the $c(i)$th predicted range (an example function in the symmetric group is defined by $c(1) = 2$, $c(2) = 3$, and $c(3) = 1$ for $N = 3$). The association loss that assumes a config c is denoted by ξ^c and encapsulates a trilateration loss, which measures the goodness of fit, and two regularization terms

$$\xi^c(x, y) = L^c_{\text{trilat.}}(x, y) + \beta_v L_{\text{velocity}}(x, y) + \beta_b L_{\text{boundary}}(x, y) \tag{2}$$

where β_v and β_b are balancing factors. The optimal config c^* is computed by solving $N!$ regularized trilateration problems (the minimization inside the parenthesis)

$$c^* = \underset{c \in \mathbf{C}}{\operatorname{argmin}} \left\{ \min_{(x,y)} \xi^c(x, y) \right\}. \tag{3}$$

a) Trilateration loss ($L^c_{trilat.}$): It ensures that the range between each reflector and a position (x, y) being optimized matches the corresponding range predicted by the neural network

$$L^c_{\text{trilat.}}(x, y) = \sum_{i \in \{1,2,\ldots,N\}} \left(\|\mathbf{P}_i - (x, y, z_{\text{fixed}})\|_2 - \hat{\mathbf{r}}_{c(i)}(t)\right)^2 \tag{4}$$

where $\mathbf{P}_i \in \mathbb{R}^3$ is the known position of the ith reflector and $\hat{\mathbf{r}}_{c(i)}(t) \in \mathbb{R}$ is the predicted range for the ith reflector, assuming config c. We assume a constant, known height, z_{fixed}.

b) Velocity loss ($L_{velocity}$): It encourages a smooth, constant velocity of the trajectory and is defined by

$$L_{\text{velocity}}(x, y) = \|\mathbf{v}(t) - \mathbf{v}(t-1)\|_2^2 \tag{5}$$

where $\mathbf{v}(\cdot) \in \mathbb{R}^2$ computes the 2-D velocity on the horizontal plane through finite difference

$$\mathbf{v}(t) = (x, y) - \left(x^*_{t-1}, y^*_{t-1}\right) \tag{6}$$

$$\mathbf{v}(t-1) = \left(x^*_{t-1}, y^*_{t-1}\right) - \left(x^*_{t-2}, y^*_{t-2}\right) \tag{7}$$

where x^* and y^* denote the predicted positions from two previous timesteps.

c) Boundary loss ($L_{boundary}$): It penalizes position predictions in the invalid region behind the reflectors and is computed by

$$L_{\text{boundary}}(x, y) = \min(0, -d(x, y)) \tag{8}$$

where $d(x, y)$ is the distance from (x, y) to the reflectors' group. As shown in Fig. 2(c), this distance is computed along the "front" direction, which is perpendicular to the red dotted line connecting the leftmost and rightmost reflectors. Positions in front of the reflectors will have positive distances, incurring zero cost, while those behind the reflectors incur positive costs.

We solve the regularized trilateration problems using L-BFGS, a memory-efficient variant of the quasi-Newton BFGS method. Then, we select the best configuration, which has the lowest $\xi^{c^*}(x, y)$, and use its optimized position (x, y) as the output for timestep t, denoted by (x^*_t, y^*_t).

C. Neural Network Implementation Details

We performed hyperparameter grid searching for UNet and Transformer networks. We started with batch size and learning rate and then expanded to other network-specific parameters. For UNet, we tuned the number of upsampling/downsampling layers and the initial channels together, but tune the temporal window size separately.

1) Optimal Hyperparameters for UNet: Upsampling/downsampling layers of 3, initial channels of 256, batch size of 256, temporal window size of 512, and a learning rate of 0.002.

For the Transformer, we separately tuned the patch width and temporal window size, while ensuring the input and output MLPs had an equal number of parameters and were tuned concurrently. The number of encoder blocks and Transformer heads was also adjusted together, with a fixed latent size per head of 32. The size of the feedforward layer was set to 4 times that of the latent size, and the dimension of the positional encoder was fixed at 4.

2) Optimal Hyperparameters for Transformer: A batch size of 1024, learning rate of $1.5625e^{-05}$, feedforward dimension of 1024, latent dimension per head of 32 with eight heads, four encoder blocks, patch width of 4 (across the range bin dimension), positional encoding dimension of 4, and temporal window size of 64. The input and output MLPs comprise two layers with 64 nodes each and a ReLU activation function in between. The final projection network utilizes the same architecture, outputting three ranges per sample.

For optimization, we used the Adam optimizer [47], a standard optimizer for training neural networks, with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay. We did not employ data augmentation, regularization, or early stopping for range prediction. Each configuration used a single NVIDIA 2080Ti GPU to train with PyTorch version 2.0.1, CUDA 11, and Python 3.8.18.

V. Experimental Results and Discussion

This section evaluates the performance of our system. We analyze range and spatial errors, and in the ablation studies, we assess the effects of different hyperparameters, regularization, and the size of the training data.

1) *Running Time:* We recorded the elapsed time and calculated the average per timestep as we performed batch processing. The range prediction and association steps took 0.84 and 13.50 ms per timestep, respectively, achieving an algorithm speed of 69.76 Hz, suitable for real-time robot navigation.
2) *Data Split:* In our main experiments and most of our ablation studies, we performed eightfold cross validation, with each scene taking turns as the test set. We used a ratio of 6:1:1 for the training, evaluation, and testing splits. We use Scene 6 as the evaluation set in all but onefold, in which we use Scene 7.

A. Errors: Range and Positional

In this section, we examine UNet's prediction errors. We report both range and positional errors across the test sets and analyze these errors with respect to three variables: the actual range, the distance from the reflector's center, and speed.

Fig. 7 presents the results: the first three columns show the errors with respect to the three variables with red lines indicating binwise average, while the last column displays RMSE histograms. The top and bottom rows represent range and positional errors, respectively. The range and positional test errors were approximately 0.15 and 0.20 m. We discuss each variable in detail below.

1) Errors Versus Actual Range: The range errors remain relatively stable up to about 5.5 m, beyond which they marginally increase. In contrast, the positional errors remain stable up to approximately 5.0 m, but increase more sharply after that. Another feature observable in Fig. 7 is the diagonal streak of data points from the bottom left to the top right in each of the two left diagrams on the bottom row. These errors stem from an effect we call the *flipping problem*, which occurs when the output ends up on the wrong side of the reflectors' center line due to the wrong reflector-range association, which will be further discussed in Section V-B2. The actual paths are illustrated by the blue lines in the same figure.

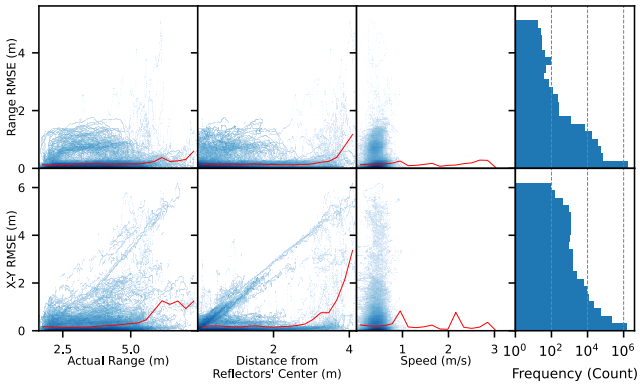


Fig. 7. UNet results: the left three columns of 2-D log histograms show how the radar's actual ranges, its offset from the reflector's center, and its motion speed impact range (top) and positional (bottom) errors. The red lines are the respective average errors. The last column displays the error distribution in log scale.

With relatively low errors in the range under 5.5 m, one suitable application could be secondary positioning for a robot arm end-effector, where the distance is adequate. To further expand the working area, one could explore techniques to join multiple workspaces together or use more reflectors to enhance accuracy at longer ranges.

2) Errors Versus Distance From Reflector's Center: In the second column, the range errors rise at a distance of about 3 m from the reflectors' center line. The positional errors increase more sharply when moving beyond 3 m.

At higher distances from the reflectors' center line, the errors from the flipping problem can contribute significantly to the average RMSE as the errors are inherently correlated with such distances. This suggests a critical factor to consider for further accuracy improvement.

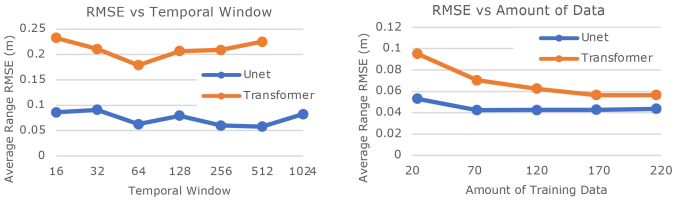
3) Errors Versus Speed: There is a slight variation in the range errors with respect to movement speed, though they do not exhibit a definitive trend. Similarly, the fluctuations in the positional errors also show no clear trend.

In this work, the maximum radar speed is 3 m/s, and the radar frame rate is set to 140 ft/s, resulting in a maximum speed per frame of approximately 0.021 m, which spans about half the width of a range bin. Under this setting, our method performs equally well with stable errors across all speeds.

4) Error Histogram: In both range and positional error histograms, the initial bins (covering from 0 up to approximately 0.25-m errors) contain significantly more data points than the second bin, as plotted on a log scale. This long-tailed distribution also suggests that addressing the relatively few high-error points could greatly enhance positional and range accuracy.

B. Ablation Studies

1) Hyperparameter: Temporal Window: In this section, we present the effects of the temporal window size, a hyperparameter in the preprocessing step, used by both the UNet and the Transformer. We use the average range RMSE of the evaluation sets as the metric.



(a) (b)

Fig. 8. Average range RMSE versus (a) temporal window size and (b) training data size.

TABLE III
AVERAGE POSITIONAL RMSES FOR REGULARIZATION SETTINGS

β_v	β_b	**Average Eval RMSE ↓**
0	0	0.2483
0	10	0.2306
0.0003	0	0.2074
0.0003	10	**0.1898**

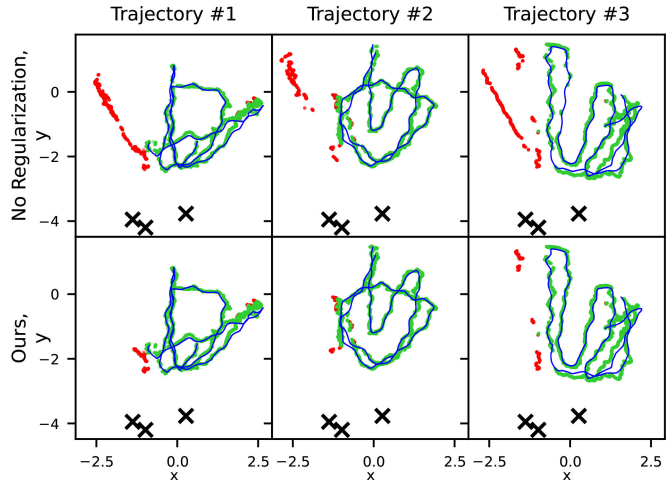


Fig. 9. Predicted trajectories without regularization (top row) and with our proposed regularization $\beta_v = 0.0003$ and $\beta_b = 10$ (bottom row). Ground-truth positions are colored blue, and predicted positions are colored green or red depending on whether their errors are low ($\leq$0.2 m) or high ($>$0.2 m). The top row clearly illustrates the flipping problem, which occurs when position optimization settles on a local minimum on the wrong side of the reflectors' center line.

Fig. 8(a) shows the range error associated with each temporal window size. The optimal temporal windows for the UNet and Transformer are 512 and 64, respectively. These values indicate the extent to which past data are needed to predict output ranges. The larger depth requirement of the UNet suggests it relies more heavily on historical data for accuracy.

2) Association Regularization: With the optimal regularization parameters, β_v and β_b, in (2), we explore the effects of the regularizing terms on the positional error. We use the average positional RMSE of the evaluation sets as the metric and compare cases with each parameter toggled on or off.

Table III displays errors from various regularization terms. Each column in Fig. 9 represents a different test trajectory, with the top row showing effects without regularization, and the bottom with optimal regularization ($\beta_v = 0.0003$ and $\beta_b = 10$). The ground-truth trajectories are illustrated in blue, and our predicted trajectories in green if the error is under 0.2 m or red if the error is at least 0.2 m, indicating the flipping problem. The reflectors are marked by crosses. With the optimal regularization, the flipping problem is highly reduced, resulting in far fewer points with high errors.

3) Training Data Size: In this section, we examine the effects of training data size on the two models. Since training data size is not a hyperparameter and generally more is better, we report our performance using a fixed, unseen test set, rather than the evaluation set previously used for hyperparameter tuning. This test set was constructed from 24 random heldout sessions, and the training set varied across five sizes: 24, 72, 120, 168, and 216 sessions. Each model, the UNet and the Transformer, was trained five times, once for each training set size.

Fig. 8(b) displays the results, showing greater error reduction in the Transformer network as the training data increases. While UNet's error plateaus from a training size of 72 sessions, adding more data slightly increases its error. This may be attributed to the model beginning to overfit the training data. The Transformer network approaches a plateau upon reaching the maximum training size of 168 sessions. Across all training sizes, UNet consistently yields lower errors than the Transformer. However, the Transformer's error decreases more sharply with increases in training size compared with UNet, indicating a greater dependency on data volume.

These findings suggest that UNet is better suited for our current dataset, as it already achieves lower errors without using all available data for training. The Transformer, on the other hand, could be advantageous with a larger dataset.

4) Test Scenes With Clutter or Interference: In this section, we test our performance when the scene contains clutter that may occlude the line of sight from the radar to the reflectors, as well as when there are multiple radars in the scene, which can cause interference. To test these effects, we conducted a controlled experiment in which we kept the radar trajectory and scene constant and only varied the levels of clutter or radar interference in each capture. In particular, the user with the radar attached was instructed to walk along a predefined trapezoidal path, depicted by the yellow dashed line in Fig. 10(a), reaching each corner at a controlled timing guided by audible cues. The clutter (a chair) or interference (another transmitting radar) was placed at either Position 1, 2, or 3. Various clutter settings of Scene 1 are shown in Fig. 10(c)–(e). Scenes 2 and 3 are shown in Fig. 10(f) and (g).

For this test, we gathered 24 test captures from three different scenes, each across eight settings and lasting approximately 60 s (see Table IV for setting descriptions). Setting BL is the baseline with no additional clutter. Settings C1–C3 represent **clutter** from a chair at different positions and angles. Settings I1–I4 represent **interference** from another radar placed at different positions and angles. The experimental results are shown in Table IV and Fig. 10(b).

a) Clutter settings: Compared with the baseline (BL), localization accuracy was degraded in Settings C3 and C2. Setting C1 also shows a slight increase in RMSE. This degradation correlates with the extent to which obstacles obstruct the sensor's view, ranging from 0.004 m from slight occlusion to 0.097 m from heavy occlusion.

b) Interference settings: As shown by the patterned bars in Fig. 10(b), Setting I1 exhibits an RMSE similar to the BL,

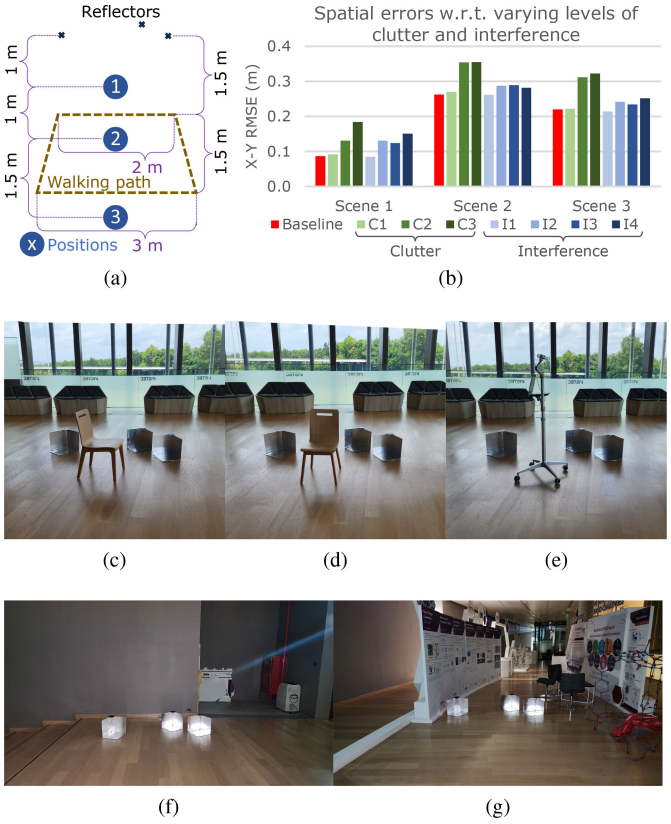


Fig. 10. (a) Test setup: the subject walks along the yellow walking path, taking 22 seconds per round. (b) Chart showing the errors from the clutter/interference experiment. (c)–(e) Photographs from Scene 1 at position 1, (c) chair facing sideways, (d) chair facing backward, and (e) second radar facing reflectors. (f) and (g) Scenes 2 and 3.

TABLE IV
SPATIAL RMSES FOR SETTINGS WITH DIFFERENT LEVELS OF CLUTTER AND INTERFERENCE. REFER TO FIG. 10(a) FOR THE SETUP

Setting	Description	RMSE (m)		
		Sc. 1	Sc. 2	Sc. 3
BL	Baseline without clutter / interference	0.0869	0.2628	0.2197
C1	Chair at ❷ facing backward	0.0915	0.2699	0.2216
C2	Chair at ❶ facing backward	0.1310	0.3544	0.3118
C3	Chair at ❶ facing sideway	0.1840	0.3548	0.3223
I1	2nd Radar at ❸ facing reflectors	0.0850	0.2618	0.2141
I2	2nd Radar at ❶ facing sideway	0.1307	0.2878	0.2416
I3	2nd Radar at ❶ facing the main radar	0.1246	0.2895	0.2345
I4	2nd Radar at ❶ facing reflectors	0.1511	0.2820	0.2516

indicating little influence from the second radar in this setting. Settings I2, I3, and I4 display slightly higher RMSEs, which are comparable to each other. These results demonstrate the minimal impact of the second radar (0.003–0.038 m), especially when compared with the clutter settings.

c) Across scenes: We observed that the overall RMSEs from Scenes 2 and 3 are significantly higher than in Scene 1, correlating with the amount of environmental clutter present in each scene.

VI. CONCLUSION AND FUTURE WORK

We present a localization approach for a commercial UWB radar sensor using simple passive reflectors. Unlike previous methods, our setup is significantly simpler and more cost-effective. The proposed system contains three steps: preprocessing, range prediction, and reflector-range association. Two neural network architectures were explored for range processing, achieving a positional error of approximately 0.2 m. This accuracy enables human or robot tracking applications at a lower cost. Our dataset will be publicly released to facilitate further research. Future work includes exploring multiple radar sensors, investigating the effects of using different numbers of reflectors or other types of radar sensors and improving robustness under heavily cluttered environments.

ACKNOWLEDGMENT

The authors would like to thank Dr. Theerawit Wilaiprasitporn for his guidance and for supplying the radar sensor critical to our research.

REFERENCES

[1] Y. Almalioglu, M. Turan, C. X. Lu, N. Trigoni, and A. Markham, "Milli-RIO: Ego-motion estimation with low-cost millimetre-wave radar," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3314–3323, Feb. 2021.

[2] Z. Peng and C. Li, "Portable microwave radar systems for short-range localization and life tracking: A review," *Sensors*, vol. 19, no. 5, p. 1136, Mar. 2019.

[3] M. Mercuri et al., "2-D localization, angular separation and vital signs monitoring using a SISO FMCW radar for smart long-term health monitoring environments," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11065–11077, Jul. 2021.

[4] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *Proc. 11th USENIX Conf. Netw. Syst. Design Implement.*, Apr. 2014, pp. 317–329.

[5] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via RF body reflections," in *Proc. 12th USENIX Symp. Networked Syst. Des. Implement.*, 2015, pp. 279–292.

[6] Z. Fang et al., "A CMOS-integrated radar-assisted cognitive sensing platform for seamless human–robot interactions," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–4.

[7] S. Lim and S. Lee, "Hough transform based ego-velocity estimation in automotive radar system," *Electron. Lett.*, vol. 57, no. 2, pp. 80–82, Jan. 2021.

[8] C. D. Monaco and S. N. Brennan, "RADARODO: Ego-motion estimation from Doppler and spatial data in RADAR images," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 3, pp. 475–484, Sep. 2020.

[9] J. Shin and S. Lee, "Estimation of ego-position using frequency-modulated continuous wave radar sensor data," *J. Korean Inst. Electromagn. Eng. Sci.*, vol. 33, no. 5, pp. 398–402, May 2022.

[10] D. Yang, Z. Zhu, J. Zhang, and B. Liang, "The overview of human localization and vital sign signal measurement using handheld IR-UWB through-wall radar," *Sensors*, vol. 21, no. 2, p. 402, Jan. 2021.

[11] D. T. A. Nguyen, H.-G. Lee, E.-R. Jeong, H. L. Lee, and J. Joung, "Deep learning-based localization for UWB systems," *Electronics*, vol. 9, no. 10, p. 1712, Oct. 2020.

[12] D. Tan Anh Nguyen, H.-G. Lee, J. Joung, and E.-R. Jeong, "Convolutional neural network-based UWB system localization," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 488–490.

[13] D. Kocur, M. Švecová, and J. Rovnáková, "Through-the-Wall localization of a moving target by two independent ultra wideband (UWB) radar systems," *Sensors*, vol. 13, no. 9, pp. 11969–11997, Sep. 2013.

[14] G. Li, Y. Ge, Y. Wang, Q. Chen, and G. Wang, "Detection of human breathing in non-line-of-sight region by using mmWave FMCW radar," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[15] M. Mercuri, E. Arnieri, R. De Marco, P. Veltri, F. Crupi, and L. Boccia, "Reconfigurable intelligent surface-aided indoor radar monitoring: A feasibility study," *IEEE J. Electromagn., RF Microw. Med. Biol.*, vol. 7, no. 4, pp. 354–364, Dec. 2023.

[16] A. Poulose and D. S. Han, "UWB indoor localization using deep learning LSTM networks," *Appl. Sci.*, vol. 10, no. 18, p. 6290, Sep. 2020.

[17] G. Schouten and J. Steckel, "RadarSLAM: Biomimetic SLAM using ultra-wideband pulse-echo radar," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–8.

[18] J. Steckel and H. Peremans, "BatSLAM: Simultaneous localization and mapping using biomimetic sonar," *PLoS One*, vol. 8, no. 1, Jan. 2013, Art. no. e54076, doi: 10.1371/JOURNAL.PONE.0054076.

[19] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3131–3137.

[20] S. Wehrli, D. Barras, F. Ellinger, and H. Jackel, "Integrated active pulsed reflector for FMCW radar localization," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2009, pp. 81–84.

[21] L.-X. Chuo, Z. Luo, D. Sylvester, D. Blaauw, and H.-S. Kim, "RF-Echo: A non-line-of-sight indoor localization system using a low-power active RF reflector ASIC tag," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Jul. 2017, pp. 222–234.

[22] S. Hansen, C. Bredendiek, G. Briese, and N. Pohl, "A compact harmonic radar system with active tags at 61/122 GHz ISM band in SiGe BiCMOS for precise localization," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 1, pp. 906–915, Jan. 2021.

[23] N. Decarli, F. Guidi, and D. Dardari, "Passive UWB RFID for tag localization: Architectures and design," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1385–1397, Mar. 2016.

[24] A. Ramos, A. Lazaro, R. Villarino, and D. Girbau, "Time-domain UWB RFID tags for smart floor applications," in *Proc. IEEE RFID Technol. Appl. Conf. (RFID-TA)*, Sep. 2014, pp. 165–169.

[25] L. Taponecco, A. A. D'Amico, and U. Mengali, "Joint TOA and AOA estimation for UWB localization applications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, pp. 2207–2217, Jul. 2011.

[26] K. Yu, K. Wen, Y. Li, S. Zhang, and K. Zhang, "A novel NLOS mitigation algorithm for UWB localization in harsh indoor environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 686–699, Jan. 2019.

[27] L. Zwirello, T. Schipper, M. Harter, and T. Zwick, "UWB localization system for indoor applications: Concept, realization and analysis," *J. Electr. Comput. Eng.*, vol. 2012, no. 1, pp. 1–24, Jan. 2012.

[28] S. P. Rana, M. Dey, H. U. Siddiqui, G. Tiberi, M. Ghavami, and S. Dudley, "UWB localization employing supervised learning method," in *Proc. IEEE 17th Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Sep. 2017, pp. 1–5.

[29] C.-H. Hsieh, J.-Y. Chen, and B.-H. Nien, "Deep learning-based indoor localization using received signal strength and channel state information," *IEEE Access*, vol. 7, pp. 33256–33267, 2019.

[30] Z. E. Khatab, A. Hajihoseini, and S. A. Ghorashi, "A fingerprint method for indoor localization using autoencoder based deep extreme learning machine," *IEEE Sensors Lett.*, vol. 2, no. 1, pp. 1–4, Mar. 2018.

[31] R. S. Sinha and S.-H. Hwang, "Comparison of CNN applications for RSSI-based fingerprint indoor localization," *Electronics*, vol. 8, no. 9, p. 989, Sep. 2019.

[32] G. Félix, M. Siller, and E. N. Álvarez, "A fingerprinting indoor localization algorithm based deep learning," in *Proc. 8th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2016, pp. 1006–1011.

[33] Z. Chen, H. Zou, J. Yang, H. Jiang, and L. Xie, "WiFi fingerprinting indoor localization using local feature-based deep LSTM," *IEEE Syst. J.*, vol. 14, no. 2, pp. 3001–3010, Jun. 2020.

[34] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate RSSI indoor localization," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019.

[35] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, Jan. 2020.

[36] J. Joung, S. Jung, S. Chung, and E. Jeong, "CNN-based Tx–Rx distance estimation for UWB system localisation," *Electron. Lett.*, vol. 55, no. 17, pp. 938–940, Aug. 2019.

[37] D. Uzunidis, E. Margaritis, C. Chatzigeorgiou, C. Z. Patrikakis, and S. A. Mitilineos, "A dataset for aftermath victim detection behind walls or obstacles using an UWB radar sensor," in *Proc. 12th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, Jun. 2023, pp. 1–5.

[38] J. Maitre, K. Bouchard, C. Bertuglia, and S. Gaboury, "Recognizing activities of daily living from UWB radars and deep learning," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113994.

[39] S. Ahmed, D. Wang, J. Park, and S. H. Cho, "UWB-gestures, a public dataset of dynamic hand gestures acquired using impulse radar sensors," *Sci. Data*, vol. 8, no. 1, p. 102, Apr. 2021.

[40] Z. Zhengliang, Y. Degui, Z. Junchao, and T. Feng, "Dataset of human motion status using IR-UWB through-wall radar," *J. Syst. Eng. Electron.*, vol. 32, no. 5, pp. 1083–1096, Oct. 2021.

[41] (2017). *X4m03 Datasheet*. [Online]. Available: https://laonuri.techyneeti.com/wp-content/uploads/2019/02/X4M03_DATASHEET.pdf

[42] A. A. Abbas, M. El-Absi, A. Abuelhaija, K. Solbach, and T. Kaiser, "Corner reflector tag with RCS frequency coding by dielectric resonators," *IET Microw., Antennas Propag.*, vol. 15, no. 6, pp. 560–570, May 2021.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-assisted Intervention—MICCAI*. Cham, Switzerland: Springer, 2015.

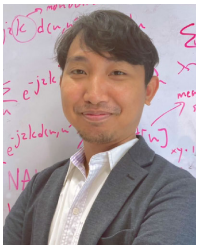
[44] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–24.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[46] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Palakon Kotchapansompote received the B.Eng. (summa cum laude) degree in EE from Thammasat University, Pathum Thani, Thailand, in 2008, and the M.Eng. degree from the University of Tokyo, Tokyo, Japan, in 2012. After several years in the industry, he is now pursuing the Ph.D. degree in machine learning applications in radar systems with the School of Information Science and Technology (IST), Vidyasirimedhi Institute of Science and Technology (VISTEC), Rayong, Thailand.



Nakorn Kumchaiseemak (Member, IEEE) received the B.Sc. degree from the King Mongkut's University of Technology, Bangkok, Thailand, in 2014, and the M.S. degree in physics from Kasetsart University, Bangkok, in 2017. He is now pursuing the Ph.D. degree with the School of Information Science and Technology, VISTEC, Rayong, Thailand.

He is currently with the Microwave Sensing, Signals and Systems Group, TU Delft, CD, Delft, The Netherlands. His research interests include the intersection of radar signal processing, human–computer interaction, and deep learning.

Kamol Kaemarungsi (Senior Member, IEEE) received the bachelor's degree in electronics engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1994, the M.Sc. degree from the University of Colorado at Boulder, Boulder, CO, USA, in 1999, and the Ph.D. degree from the University of Pittsburgh, Pittsburgh, PA, USA, in 2005.

He is currently the Research Director of the Communications and Network Research Group (CNWRG) under the National Electronics and Computer Technology Center (NECTEC), Pathum Thani, Thailand. His research interests include ground penetrating radar, ultrawideband sensors, wireless sensor networks, and indoor positioning systems.

Supasorn Suwajanakorn received the B.Eng. degree from Cornell University, Ithaca, NY, USA, in 2011, and the Ph.D. degree from the University of Washington, Seattle, WA, USA, in 2017.

He is currently a Lecturer with the School of Information Science and Technology (IST), Vidyasirimedhi Institute of Science and Technology (VISTEC), Rayong, Thailand. His research interests include the intersection of computer vision, deep learning, and computer graphics.