DeepCIR: Insights into CIR-based Data-driven UWB Error Mitigation

Vu Tran* Zhuangzhuang Dai* Niki Trigoni* Andrew Markham*

***Abstract*— Ultra-Wide-Band (UWB) ranging sensors have been widely adopted for robotic navigation thanks to their extremely high bandwidth and hence high resolution. However, off-the-shelf devices may output ranges with significant errors in cluttered, severe non-line-of-sight (NLOS) environments. Recently, neural networks have been actively studied to improve the ranging accuracy of UWB sensors using the channel-impulse-response (CIR) as input. However, previous works have not systematically evaluated the efficacy of various packet types and their possible combinations in a two-way-ranging transaction, including *poll*, *response* and *final* packets. In this paper, we firstly investigate the utility of different packet types and their combinations when used as input for a neural network. Secondly, we propose two novel data-driven approaches, namely FMCIR and WMCIR, that leverage two-sided CIRs for efficient UWB error mitigation. Our approaches outperform state-of-the-art by a significant margin, further reducing range errors up to 45%. Finally, we create and release a dataset of transaction-level synchronized CIRs (each sample consists of the CIR of the *poll*, *response* and *final* packets), which will enable further studies in this area.**

I. Introduction

Ultra-Wide-Band (UWB) ranging sensors have proven to be a robust technology for indoor localization and tracking [1], [2], [3]. Since UWB sensors use RF pulses to measure the distance to another, they can work through opaque materials that usually occlude optical signals (e.g. laser). By using an extremely high bandwidth ($\geq$ 500 MHz), much higher than narrow-band counterparts (e.g. WiFi, Bluetooth), UWB sensors can achieve a very high time and space resolution [4] and thus high accuracy. These properties make UWB sensors extremely useful for robotic navigation as robots can use them to measure ranges to reference points (also equipped with UWB sensors). UWB sensors have been used to either localize a robot directly or cure the drift caused by unavoidable accumulative errors [5], [6], [7], [8], [9], [10].

Despite being capable of fine ranging, UWB sensors may fail to provide accurate ranges in adverse environments. The widely used method to obtain range is to measure the time of flight (TOF) of RF pulses, which may arrive at a receiver via different travelling paths. The receiver detects the timing of the first arriving signal using a leading-edge detection (LDE) algorithm [11] and thus largely avoids the effects of delayed signals existing in a multi-path environment. In severe Non-Line-Of-Sight (NLOS) channels, the direct path may, however, become too weak to be detected using heuristic tuning, e.g. choice of thresholds. Hence, UWB error mitigation has been intensively studied to improve UWB ranging under challenging conditions.

*The authors are with the Department of Computer Science, the University of Oxford firstname.lastname@cs.ox.ac.uk

Recently, neural networks have drawn considerable interest as a data-driven approach for UWB error mitigation [12], [13], [14], [15], [16]. A neural network usually takes as input a single snapshot of the channel impulse responses (CIR), which captures the channel properties, and outputs an estimation of the error made by the built-in algorithm. The estimated error is then subtracted from the raw measurement to obtain the corrected range. Neural networks have shown to be more robust than hand-craft heuristics in detecting the direct path (usually weak) in cluttered, NLOS environments.

Though promising, state-of-the-art (SOTA) approaches do not provide insights into the efficacy of different packets. Given the prevalence of the two-way ranging scheme that incorporates multiple packets in each transaction, we shall show in section V that using an inappropriate packet as input may result in significantly lower mitigation performance. More importantly, taking as input a single CIR is sub-optimal because of time-variations of real-world RF channels (with sub-millisecond coherence time [17]). To gain more insights, we create a transaction-level synchronized CIR dataset that enables us to study the impact of various CIR combinations. Primarily, we propose two neural networks that learn from two-sided CIRs effectively, resulting in significant improvements over the single-CIR approaches used in previous works. In summary, we make the following contributions:

- We systematically investigate and evaluate the efficacy of various combinations of CIRs as input for a neural network. Our studies provide insights into efficient uses of CIR for data-driven UWB error mitigation.
- We propose two novel approaches, namely FMCIR and WMCIR, for UWB error mitigation in challenging environments. Our approaches enable a neural network to exploit multiple CIR snapshots in a transaction to mitigate ranging errors better, outperforming the state-of-the-art approach by a significant margin.
- We create a transaction-level synchronized CIR dataset[1] to enable further studies. Our dataset includes more than 120k ranging transactions recorded in a variety of environments and conditions. Unlike previous work, each sample in our dataset consists of synchronized CIRs and diagnostic details of all 3 packets: *poll*, *response*, and *final*, in a regular transaction.
- Finally, we have evaluated FMCIR and WMCIR under various conditions, including LOS/NLOS channels using stationary/moving tags. We find FMCIR and WMCIR work well in unseen environments and efficiently mitigate drifting in a downstream odometry task.

[1]Download dataset and code: https://github.com/vutran86/DeepCIR

II. Background & Related Works

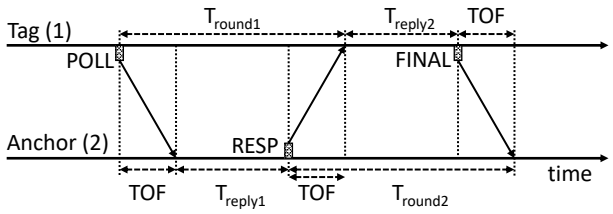


Fig. 1. Two-way-ranging protocol e.g. as used in Decawave sensors. Each ranging transaction consists of three packets, namely: *poll*, *response* and *final*. The Tx and Rx timestamps are used to calculate the TOF and mitigate the impact of clock drift.

UWB ranging has been studied intensively for years and achieved great advances by using either Angle-of-Arrival (AoA) [3] or TOF [18]. DW1000 sensor [18], a typical UWB sensor that uses TOF to measure range, is widely used in research studies [12], [13], [14], [15], [16] because of its high accuracy as well as portability. Essentially, TOF can be measured if the transmitter and the receiver are perfectly synchronized and can record the timestamps of transmitted and received packets accurately. However, even near-perfect synchronization is extremely difficult in wireless channels. Two-Way-Ranging (TWR) was proposed to estimate TOF without node synchronization [19]. However, TWR still suffers heavily from clock drift and frequency drift. Double-sided two-way-ranging (DS-TWR) is an alternative ranging method that can reduce the clock/frequency drift but requires at least three packets to estimate TOF [20]. These packets in a DS-TWR scheme of DW1000 are illustrated in Figure 1. Firstly the tag sends a *poll* packet. The anchors that receive the *poll* packet will take turns transmitting a *response* packet. The tag that receives *response* packets from anchors then transmits a *final* packet to all the anchors. The sensor can accurately record the transmission and reception timestamps and compute the TOF using the following equation [20].

$$TOF = \frac{T_{round1} \times T_{round2} - T_{reply1} \times T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} \quad (1)$$

While the transmission timestamp is independent of the wireless channel, the reception timestamp greatly depends on the wireless channel. The reception timestamp is marked at the moment the receiver detects the earliest arriving signals of a packet, or first-path-index (FPI). Under challenging conditions, a sensor may inaccurately estimate the FPI and thus the range. Though UWB error mitigation has been intensively studied, mitigation under severely cluttered/NLOS conditions remains a challenging problem.

A. Related works

UWB ranging has been widely studied for robotic applications thanks to its fine-grained ranges. Zheng *et al.* [6] utilize a UWB-mesh network for mobile robot localization. Tiemann *et al.* [7] use UWB sensors along with a camera to enhance monocular SLAM for a UAV in indoor environments. UWB sensors can be combined with Lidar to enable range-only SLAM [10], or with IMU sensors to improve the localization of a UAV [8]. Cao *et al.* [9] applied UWB ranging for relative localization among multiple robots.

Despite its high ranging resolution, UWB sensors still suffer from heavily cluttered and/or NLOS environments. This problem has been intensively studied [21], [22], and is still actively studied using data-driven methods [23], [24]. Many studies attempt to solve the problem by (1) classifying LOS/NLOS condition from the signal, and (2) mitigating the erroneous NLOS ranging transactions. Traditional algorithms based on hand-crafted features such as first-path detection [22] or statistical thresholding [23], perform unreliably in different NLOS scenarios. Non-parametric data-driven methods, such as LS-SVM [21] or Gaussian processes [25], only provide limited gains in real-world NLOS environments.

Recently, neural networks have emerged as a promising UWB error mitigation method. Jiang *et al.* [24] propose to use a CNN (Convolutional Neural Network)-LSTM (Long Short-Term Memory) network to classify LOS/NLOS based on CIR. Hsiao *et al.* [26] use synthetic CIRs to train a neural network to estimate FPI. Bregar *et al.* [12] propose a simple CNN to detect NLOS transactions and estimate the ranging errors. Fontaine *et al.* [13] propose an auto-encoder to learn the representation of the CIR, and use this representation to estimate the error via a CNN. Ridolfi *et al.* [14] propose a neural network to correct UWB ranges for anchor self-calibration. Nonetheless, there is a lack of insightful investigations of efficient uses of CIRs, given that the commonly used UWB sensors require different packets from both sides for a range measurement.

B. Need for systematic studies of efficient uses of CIR

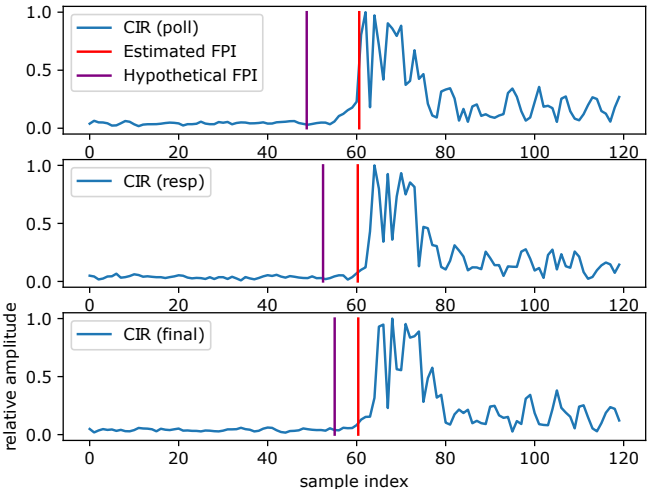


Fig. 2. CIR buffers of *poll*, *response* and *final* packets in the same transaction (resolution $\approx 1ns/sample$). Using the FPI estimated by the sensor results in an error of 2.8 m, while using the imaginary/hypothetical FPI (precede a probable "peak") would result in an error of 0.0 m.

Recent data-driven mitigation approaches use only a single CIR snapshot as input, whereas a typical ranging transaction consists of three packets, including *poll*, *response*, and *final*. Under perfect conditions, one can assume a reciprocal, stationary channel and that the CIRs of the three packets remain almost identical. However, we observe that this assumption

TABLE I
UWB CIR DATASETS

Dataset	Samples	Scenarios		Synchronization
		LOS	NLOS	
Fontaine [13]	28k	1	1	No
Bregar [12]	42k	7	7	No
Usman [27]	15k	2	2	No
Kram [28]	50k	1	2	No
Ours	120k	3	3	Yes (3-CIR)

does not hold practically, and even a slight variation may create a large offset in the estimated FPI, which translates into a significant range error. Figure 2 plots the CIRs of the *poll*, *response* and *final* packets under a challenging NLOS condition. The sensor-estimated FPIs (red lines) mark the timestamps used to calculate the TOF, which results in a range error of 2.8 m. Interestingly, there is a small hump immediately before the FPI, which is only conspicuous in the *resp* CIR. Assuming an imaginary/hypothetical FPI right before the hump would result in an error of 0.0 m. We note that the TOF can be re-calculated using timestamps & FPI values (using Equation 1) stored in each transaction in the dataset. Another interesting observation is that protocol-specific properties of UWB ranges may also define the efficacy of different CIRs when used as input for a neural network. By expanding Equation 1 to include error terms in reception times (for space limitation, we do not show the expanded equation), the contributions of FPI errors of *poll*, *response* and *final* packets into the transaction error are proportional to T_{reply2}, $T_{reply1} + T_{round2}$, and T_{round1}, respectively. Given $TOF << T_{roundi}$ and $TOF << T_{replyi}$, the weights can be approximately computed from the slot times in the protocol used in TREK1000 devices. For simplicity, we empirically found these weights are approximately 0.42, 0.5 and 0.08 respectively. This example suggests that having access to multiple CIR snapshots in a transaction could enhance the ability to detect the correct first peak, which in this case almost disappears within the noise floor. We particularly note that this practice does not require any additional ranging transactions. It only uses the available CIR information which has not been exploited in previous works.

Two questions naturally arise, to which answers are as yet unknown: (1) which of the three packets (*poll*, *response*, *final*) is the most effective as an input for a neural network? (2) Can combining multiple CIR snapshots in a transaction improve the error mitigation performance? Unfortunately, we note that current published datasets do not include transaction-level synchronized CIRs to enable detailed studies of specific packet types as well as their combinations.

C. UWB Datasets

Fontaine *et al.* [13] released a dataset of mixed LOS and NLOS UWB measurements with 28k samples. However, the measurement campaign was conducted in one scenario only with 23 fixed UWB agent positions. Furthermore, there are more LOS samples than NLOS ones. Bregar *et al.* [12] release a dataset of 42k samples with evenly splitted LOS and NLOS measurements in seven different indoor environments.

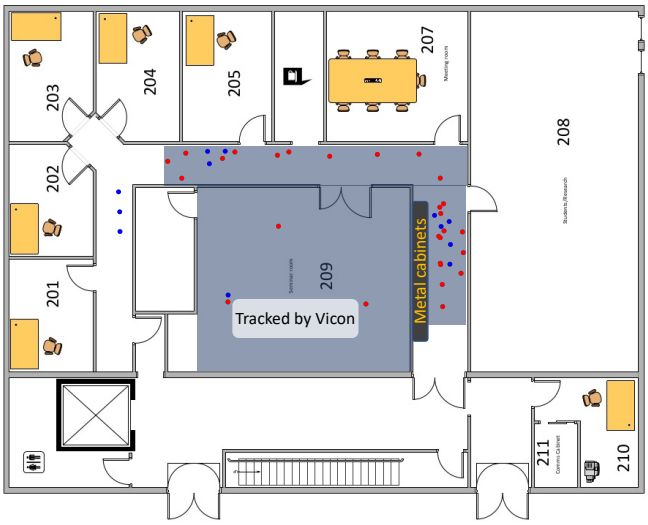


Fig. 3. Floorplan and anchor positions for data collection in an office environment. Training anchor positions are marked with *red* dots, while *blue* dots show the anchor positions used in test set. The trajectories of the mobile tag cover most of the Vicon-tracked areas (shaded).

Additionally, in [13] or [12] measurements were taken at fixed reference points which restricts the diversity of channel conditions. Usman *et al.* [27] publish data collected from UWB agents held by a walking person and on an automated guided vehicle. Kram *et al.* [28] collected 50k data from three distinct environments. However, none of the above datasets record synchronized three-way CIRs. This prompted us to create a new dataset to foster further investigations of UWB error mitigation. Details of our dataset are expanded on in Section III, while Table I provides an overview of our dataset compared with existing ones.

III. TRANSACTION-LEVEL SYNCHRONIZED CIR DATASET

This section presents details of our dataset and the procedure to obtain transaction-level synchronized CIRs, which later enable us to systematically evaluate different CIRs and their combinations as input of a neural network.

A. Data collection

We used the Trek1000 evaluation kit [19], which includes a widely used DW1000 sensor in each device, for data collection. We recorded our dataset in an office building environment, within and around a seminar room which is instrumented with a Vicon motion capture system [29]. The Vicon system can provide near real-time location data (with sub-millimeter accuracy) used as ground-truth for our dataset. We placed most of the anchors in the Vicon-tracked areas. The anchor positions outside the Vicon-tracked area were manually measured. The tags were either placed on a tripod (≈1.2m height), or held by a person who moved around in the Vicon-tracked area. Figure 3 plots the floorplan of the office as well as the deployed positions of UWB anchors. *The tag locations are not plotted as we recorded around 50K samples using a hand-held tag moving in arbitrary trajectories that cover most of the Vicon-tracked areas (shaded).* The tag trajectories cover both LOS and NLOS

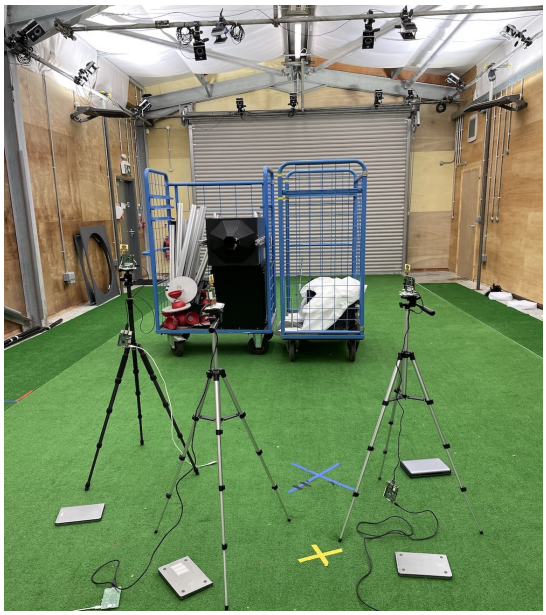


Fig. 4. Data collection settings in a flight hall. The two metal racks filled with metal objects create NLOS conditions. The metal frames on the walls functions as reflectors. The tag and the anchors are on the opposite sides of the metal racks.

regions. The *red* dots represent anchor positions used in the training set, and the *blue* dots represent the anchor positions used in the test set. We recorded the training set and test set in multiple sessions, each of which took up three distinct anchor positions. We used more anchor positions in the hallway as they conveniently created more NLOS transactions when the tag was moving in the Vicon-tracked area.

To synchronize the packets recorded separately in the tag and the anchors, we modified the firmware of the TREK100 boards to record diagnostic details (e.g., CIR, FPI, range), and especially the sequence number of different packets (*poll*, *response*, *final*) in ranging transactions. The sensors operated at a frequency of 4 GHz, with a data rate of 110 kbps. We configured the TREK1000 boards to record 120 samples (centered around the FPI) of each 992-sample CIR buffer due to its limited processing capability. As discussed in section V, 64-sample CIRs are sufficient for a neural network, and the samples that are too far from the FPI do not contain useful information. During the experiments, each sensor (both tag and anchors) was connected to a Raspberry Pi 3 board which reads and stores sensor data locally along with the timestamps. We first synchronize the clock of Raspberry Pi boards with a central server via WiFi. In each ranging transaction, the tag node initiates the transaction with an 8-bit sequence number. We set the ranging rate to 2.5 Hz (to accommodate data logging time). Consequently, as long as the clock in each Raspberry Pi board does not drift more than 50 seconds during the experiment session (which we never observed), we can be assured that the CIRs can be matched offline after the experiment using the recorded timestamps and sequence numbers. Each *synchronized* sample contains the logged data (e.g., CIR, range, FPI) of three packets: *poll*, *response*, and *final*. In total, we recorded 33K samples for

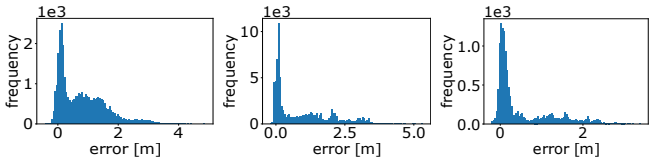


Fig. 5. Histogram of range errors in (a) training set (b) test set (c) additional test set recorded in a flight hall.

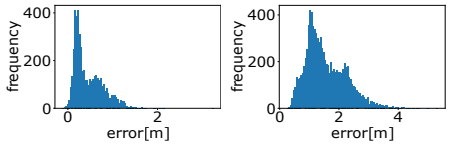


Fig. 6. Histogram of range errors under (a) LOS and mild NLOS (one wall separation) conditions (b) severe NLOS (one wall + one metal cabinet or two walls + storeroom objects)

the training set, which is split into 25K samples for training and 8.1K samples for validation. The test set includes 75.2K samples.

To assess the generalization of neural networks, we recorded another test set in a totally different environment. Figure 4 shows the flight hall in a bird flight research center where we record our additional test set. Most of the space is tracked by a Vicon motion capture system. There were also two metal racks filled with metal objects that served as obstacles that blocked the direct path. The walls also include metal frames which are a rich source of reflections. We recorded the data with nine different anchor positions, while the tag was attached to a tripod and placed at different locations on the opposite side of the metal racks. The tag was intermittently moved to various positions in three trajectories to capture both stationary and dynamic conditions. This additional test-set includes 12.7K samples (3 CIRs). Figure 5 shows the histogram of ranging errors in the dataset.

B. Ranging errors under various conditions

We observe that existing UWB CIR datasets contain few samples with large errors ($\geq 2.0\,m$). Only Hsiao *et al.* [26] reported errors up to 4.0 m in their experiment conducted in a university campus. Such limited error ranges may lead to the model being over-fitted to the modest errors and not being able to work in unseen and challenging environments with large errors. In our settings, where UWB sensors are deployed in a dense space packed with storeroom objects, especially operated with a hand-held device, we observe a higher proportion of larger ranging errors. Figure 6 plots the histogram of range errors under LOS and mild NLOS conditions (tags and anchors are separated by a wall) and a heavy NLOS condition (tags and anchors are separated by two walls + storeroom objects or one wall + one metal cabinet + storeroom objects). In the LOS and mild NLOS circumstance, the distribution of errors is quite similar to the ones seen in existing datasets, with most of the ranging transactions achieving an error below 2 m. However, when the anchors are close to a large metal object (cabinets), and

a person is standing or moving near the anchors or the tags, we observed much larger range errors. We believe our data recorded under challenging conditions diversifies the dataset and enables a neural network to generalize to unseen environments better.

IV. *DeepCIR* APPROACHES

To fully explore the transaction-level synchronized dataset, we propose two neural network models capable of handling multiple CIR snapshots as input. For comparison, these models are both based on a base model (1-CIR) that is capable of handling one CIR. Figure 7(a) shows the architecture of 1-CIR, which is similar to the model used in [14], but with modifications to the architecture to improve its performance and stability. 1-CIR includes a feature extraction block with 6 convolutional layers and a regression block with 3 fully connected layers. We decided not to include an auto-encoder [13] because we empirically found that it does not improve the performance of the model. 1-CIR takes as input a window of a CIR buffer (centered around the sensor-estimated FPI) and outputs an estimated range error. Detailed implementation can be found in our published repository[1].

Using a neural network for error mitigation, we make the assumption, which is also inherent in prior work, that the device is able to send the CIRs to a processing node that executes the neural network, or that it can execute the neural network itself. While an anchor node may have abundant processing and energy resources for sending the CIRs or executing the neural network, the mobile tag may be limited in such resources. Considering the three packets: *poll*, *response*, *final* in a transaction, using *poll* and/or *final* CIR is more convenient because they are received at the anchors. Since *response* CIR is received at the tag, it may need to be transmitted to another processing node (e.g. an anchor node) unless the tag node is sufficiently powerful (e.g. a drone) to execute the neural network. Fortunately, we found that a neural network does not need a large CIR buffer to correct the ranges. As discussed later in section V, using a 64-sample CIR buffer, FMCIR and WMCIR can significantly reduce the range error. As a CIR buffer occupies only 256 bytes (64 16-bit integer complex numbers), it can be transmitted via a Bluetooth/WiFi interface (if the tag node has one), or even embedded into a UWB packet (DW1000 sensor supports a maximum packet length of 1023 bytes [18]). Of course larger packets incur an additional energy overhead, but measuring the energy consumption is out of the scope of this paper.

A. Fused Multi-CIR: FMCIR

The simplest way to fuse multiple CIR snapshots is to concatenate them at the input layer to create an input vector with more channels. However, this simple approach requires larger convolutional layers, which increases the computational cost and causes the model to learn unwanted spatial relations among CIRs. We note that the indices in a CIR buffer are relative to the internal state of the receiving pipeline in the sensor and not aligned among packets. For instance, if the FPI of the *poll* packet is 700 and the FPI of the *final* packet is 600, it does not mean the *final* packet TOF is 100ns shorter (sampling interval $\approx$ 1ns). Therefore, the indices among two CIRs are uncorrelated and should not be mistakenly learnt during the training phase. Even though the CIR of *poll*, *response* and *final* packets are different, their nature remains the same (channel response from a transmitter to a receiver). Therefore, we propose to use a shared feature extraction layer for all inputs and only fuse the features rather than the inputs. Figure 7(b) describes the architecture of FMCIR. In FMCIR, we use the same feature extraction layer used in 1-CIR. The feature vectors are then concatenated to form a larger input vector for the regression block. The regression block in FMCIR is labelled as *Regression-M* to show that it is different from the *Regression-1* used in 1-CIR. The FMCIR model is parameterized to support both two-CIR and three-CIR inputs.

B. Weighted Multi-CIR: WMCIR

Although FMCIR uses a shared feature extraction block, the regression block is significantly expanded due to the increase in size of the feature vector. As discussed in section II, *poll*, *response*, and *final* packets have implicit weighted contributions to the ranging errors in a two-way ranging scheme. Based on this, we propose WMCIR that integrates this prior knowledge into the network architecture, forming a hybrid between a purely neural model and a physics informed model. Figure 7(c) illustrates the WM-CIR architecture. WMCIR uses the same feature extraction block and regression block as used in 1-CIR. However, the entire sub-model is shared among three inputs (CIRs). The corresponding outputs are weighted and summed with the weights that are obtained empirically via equation 1 (refer to section II). Specifically, *w1*, *w2*, and *w3* are set to 0.42, 0.5, and 0.08, respectively. However, these weights can have different values, depending on the protocol settings (slot times), which can be pre-computed.

V. EXPERIMENTS

We conducted experiments in three distinct environments, including (1) An experiment room (which is a seminar room instrumented with a Vicon motion tracking system [29]) and its corridors and (2) a flight hall (instrumented with another Vicon motion tracking system), and (3) finally, we conducted a robotic navigation experiment in the corridors around the seminar room. We used TREK1000 evaluation boards [19] for all the experiments.

To fairly compare different approaches, we applied the same training strategy for all the experiments throughout the studies. The input CIR was trimmed to have a size of 64 samples (centered around the roughly estimated FPI). Though we empirically found this input size is reasonable for a neural network, our dataset supports up to 120-sample CIRs. We kept the complex format of the CIR samples but normalized the entire CIR buffer to have a maximum amplitude of 1.0. We used a batch size of 64, Adam optimizer with a learning rate of 0.0001, and early-stopping with a patience value of

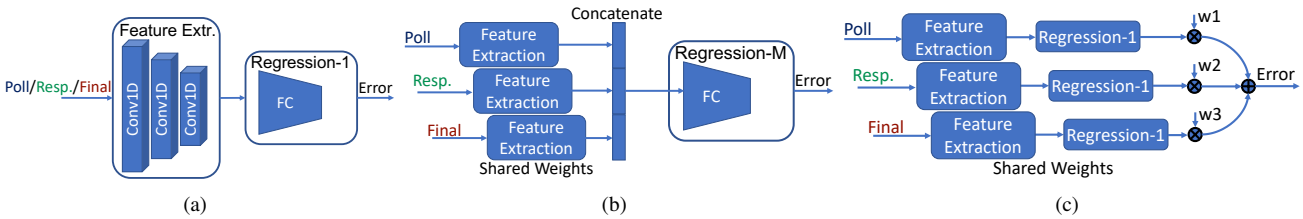


Fig. 7. Neural network architecture of (a) 1-CIR (b) FMCIR (c) WMCIR. All three architectures use the same feature extraction block, while 1-CIR and WMCIR also use the same *Regression-1* block. The estimated error (the output) is subtracted from the raw range to obtain the corrected range.

10 epochs. We also applied L2 regularization with a value of 0.0005 in all convolution and fully connected layers.

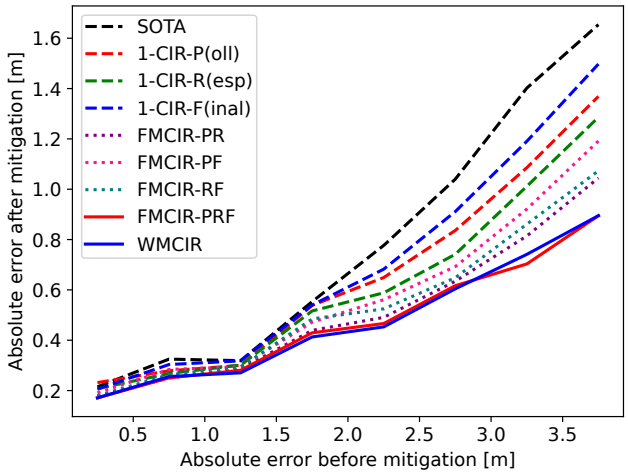


Fig. 8. Error mitigation performance using various input combinations. SOTA and 1-CIR use one CIR as input. FMCIR uses two or three CIRs as input and WMCIR uses three CIRs (PRF) as input. Definitions of suffixes: *P* for *poll*, *R* for *response*, and *F* for *final*.

A. 1-CIR performance

We first evaluate the single-CIR approaches, including the state-of-the-art model used in [14] (labelled as SOTA) and 1-CIR. SOTA proposed a neural network to correct UWB ranges for anchor self-calibration (estimating anchor positions). We replicate SOTA with the hyper-parameters reported in [14]. Though both SOTA and 1-CIR take as input a single CIR buffer, 1-CIR's architecture is empirically optimized and thus different from SOTA. We run the entire training and testing process 10 times for each approach and take the average performance to reduce the variations caused by random processes during training. Figure 8 shows the performance of single-CIR approaches in dashed lines. The horizontal axis represents the error before mitigation, which is binned into intervals of 0.5m. The vertical axis represents the error after mitigation, which is the range subtracted by the estimated error (the output of the neural network). We evaluate the SOTA using *final* packets as we believe *final* packets are the most convenient ones to record at an anchor node because the CIR of a *final* packet will overwrite the CIR buffer after each transaction. In general, the results show that using *response* packets as input significantly improves the performance over using *final* packets. The reason could be that the method (Equation 1) used by the sensor to compute ranges in a DS-TWR transaction implicitly imposes the lowest weight of the *final* packets. The SOTA performs comparably well with modest errors but considerably degrades with larger errors (challenging conditions).

B. FMCIR performance

In this experiment, we first investigate different combinations of two CIRs: {*poll*, *response*} (FMCIR-PR), {*poll*, *final*} (FMCIR-PF), {*response*, *final*} (FMCIR-RF) as the input for a neural network. We use FMCIR with two inputs as the network. The performances of the three combinations are plotted in dotted lines in Figure 8. Interestingly, using a combination of *poll* and *final* packets as input further improves the performance of the best 1-CIR approach (using *response* packet). This is extremely convenient because both *poll* and *final* packets are received at anchor nodes that typically have significant computational and energy resources (e.g., a wired computing system). We note that this approach can be used seamlessly with existing systems, without any modifications to tag firmware. However, we do note that the combinations that include CIRs from both directions (tag-to-anchor and anchor-to-tag) (i.e. {*poll*, *response*} or {*response*, *final*}) significantly improves error mitigation.

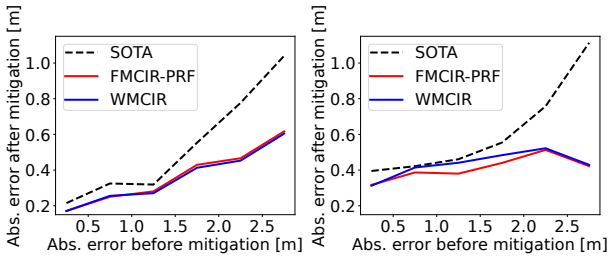
We further investigate the fusion of three CIRs {*poll*, *response*, *final*}. The solid red line in Figure 8 shows the performance of three-CIR fusion approach, the FMCIR-PRF. The results show that the fusion of three CIRs significantly improves the mitigation performance when the sensors experience large errors. Together with the 1-CIR and 2-CIR fusion, the results suggest that the channel is not stationary during a transaction (and possibly not reciprocal). Consequently, a neural network needs to take as input all three *poll*, *response* and *final* packets to achieve the highest error mitigation performance. In particular, FMCIR-PRF further reduces ranging errors up to 45% compared with the SOTA.

C. WMCIR performance

Although the FMCIR-PRF approaches achieve significantly higher performance compared to single-CIR approaches, the concatenation of feature vectors increases the number of parameters in the regression model significantly. For instance, 1-CIR contains 685,449 parameters, while FMCIR-PRF contains 2,985,481 parameters. Although these models are small in comparison with models commonly encountered in computer vision and robotics e.g. with tens or hundreds of millions of parameters, it is beneficial to limit the model size to support mobile UWB sensors such as hand-held tags or portable anchors. WMCIR reduces the model size by using a weighted sum approach, exploiting

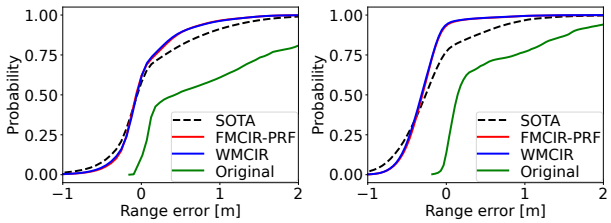
the prior knowledge of the widely used DS-TWR scheme. We note that WMCIR has the same number of trainable parameters as 1-CIR. At inference time, it does require three times as much computational cost as 1-CIR as three CIRs are processed individually. In particular, the WMCIR, FMCIR-PRF, and SOTA take 60ms, 91ms, and 26ms for an inference (average over 100 inferences) on a Raspberry Pi 3 platform, respectively. We found that this approach is extremely efficient. The solid blue line in Figure 8 shows that WMCIR achieves comparable performance with FMCIR-PRF approach whilst keeping the model significantly simpler. This also allows the tag to simply communicate a range estimate, rather than a full CIR, for range correction.

D. Generalization to unseen environment



(a) Office environment (b) Flight hall environment

Fig. 9. Error mitigation performance in (a) our office environment (seen environment) and (b) a flight hall (unseen environment).



(a) Office environment (b) Flight hall environment

Fig. 10. Cumulative distribution of range errors with and without mitigation in (a) our office environment (seen environment) and (b) a flight hall (unseen environment).

A key issue that faces data-driven techniques, and neural networks in particular, is that of over-fitting to the training set. Thus, we evaluate the generalization of the proposed approaches using the data recorded in a flight hall, which is not included during neural network training. For comparison, Figure 9(a) shows a cropped version of three approaches, including the SOTA, the FMCIR using 3 CIRs (FMCIR-PRF), and WMCIR in our office environment (inside and around a seminar room). Figure 9(b) shows the performance of the same approaches in the flight hall. It is noticeable that FMCIR and WMCIR work well with large input errors, significantly outperforming the SOTA (using 1 CIR). Interestingly, large input errors are better mitigated (compared with the office environment), while small errors (0.0 m to 0.5 m) seem not to benefit from the mitigation. More specifically, Figure 10 plots the cumulative distribution of ranging errors in (a) the office, and (b) the flight hall. Figure 10 shows that the neural networks overestimate the

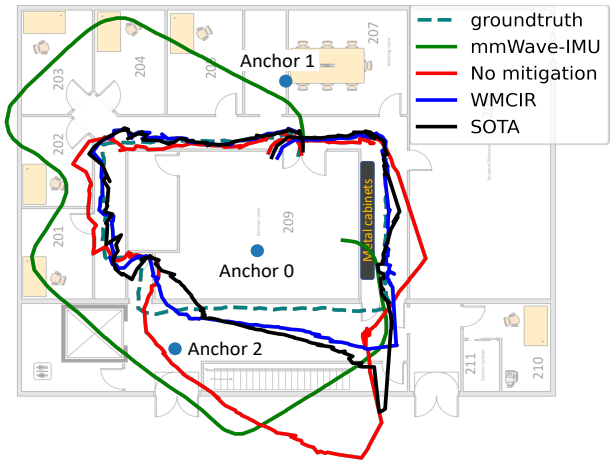


Fig. 11. Odometry correction using UWB ranges with and without error mitigation.

ranging errors more frequently in the flight hall than in the office (resulting in a higher proportion of negative errors). Nonetheless, FMCIR and WMCIR effectively mitigate most of the ranging errors, resulting in a 95% percentile of 0.7 m, a significant improvement from the SOTA (1.0 m) and the original ranging errors (2.1 m).

E. Drift mitigation in robotic navigation

Finally, we conducted an experiment to assess the usefulness of the proposed UWB error mitigation approaches for a robotic navigation task. We used the handheld odometry platform as in MilliEgo [30] to obtain the odometry data. The platform also includes a Lidar [31] to record ground-truth trajectory. The UWB tag was mounted on the Lidar, and connected to a Raspberry Pi 3 to record ranges at 2.5 Hz. The Lidar and MilliEgo were connected to a Jetson Xavier board operating at 10 Hz. The entire system moved in a trajectory around the seminar room in a counter-clockwise direction, starting from the door. Three anchors were deployed at three distinct positions, shown as three blue dots in Figure 11. All the data were recorded and processed offline after the experiment. MilliEgo [30] combines a millimeter wave radar and an Inertial Measurement Unit to estimate the trajectory of a robot or a person in visual-denied conditions. However, the predicted trajectory significantly drifts at sharp turns, resulting in a mean absolute error (MAE) of 6.2 m. We then used a particle filter along with UWB ranges (with/without mitigation) to correct the trajectory. In the particle filter, we used a Rayleigh distribution ($\sigma = 0.5\,m$) as the measurement uncertainty of raw ranges (no mitigation). Because UWB sensors measure TOF of RF waves, the errors conform to a Rayleigh distribution (long-tailed) [32], with a positive skew. After mitigation, we observed that the error distribution was more Gaussian-like, and hence used Gaussian ($\sigma = 0.5m$) as the measurement uncertainty. Figure 11 shows that WMCIR noticeably outperforms the previous SOTA. More specifically, WMCIR achieves an MAE of 0.65 m (average over the trajectory), considerably better than the SOTA (1.0 m) and no-mitigation (2.3 m).

VI. Conclusions

In this paper, we systematically studied the efficacy of different packet types (*poll*, *response* and *final*) and their combinations in DS-TWR transactions for data-driven UWB error mitigation. We showed that using bidirectional CIRs improves the error mitigation performance significantly. Our multi-CIR approaches, namely FMCIR and WMCIR, outperform the state-of-the-art approach by a significant margin. Furthermore, we created a transaction-level synchronized CIR dataset to enable further studies in this area.

Although FMCIR and WMCIR adapt well to the unseen flight hall, other environments may have inevitable disadvantages that may require fine-tuning. We note a tension between environment-agnostic and environment-specific models i.e. those that take into account anchor topology and clutter locations. In future, approaches that can combine a generic model with environment specific fine-tuning should be investigated.

Acknowledgment

This research has been financially supported by the UK Research and Innovation, the Engineering and Physical Sciences Research Council (EPSRC) via the grant ACE-OPS: From Autonomy to Cognitive assistance in Emergency OPerationS (Grant Reference: EP/S030832/1). We thank Professor Graham Taylor (University of Oxford) for his generosity in granting us access to the tracking facility. We thank Dr. Stuart Golodetz (University of Oxford) for his aid in the experiments.

References

[1] "Ultra wideband technology," https://www.zebra.com/gb/en/products/location-technologies/ultra-wideband.html, accessed: 2021-12-10.

[2] "Determine a device's relative position with uwb localization capability," https://www.nxp.com/applications/enabling-technologies/connectivity/ultra-wideband-uwb:UWB, accessed: 2021-12-10.

[3] "Ubisense real-time location system (rtls)," https://ubisense.com, accessed: 2021-12-10.

[4] M. Yavari and B. G. Nickerson, "Ultra wideband wireless positioning systems," *Dept. Faculty Comput. Sci., Univ. New Brunswick, Fredericton, NB, Canada, Tech. Rep. TR14-230*, vol. 40, 2014.

[5] Y. Xu, Y. S. Shmaliy, C. K. Ahn, G. Tian, and X. Chen, "Robust and accurate uwb-based indoor robot localisation using integrated ekf/efir filtering," *IET radar, sonar & navigation*, vol. 12, no. 7, pp. 750–756, 2018.

[6] L. Zheng, C. Ai, Z. Qi, D. Geng, Z. Feng, and H. Wang, "Research on differential positioning technology of mobile robot based on uwb," in *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering*, ser. EITCE 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 215–220. [Online]. Available: https://doi.org/10.1145/3501409.3501449

[7] J. Tiemann, A. Ramsey, and C. Wietfeld, "Enhanced uav indoor navigation through slam-augmented uwb localization," in *2018 IEEE international conference on communications workshops (ICC workshops)*. IEEE, 2018, pp. 1–6.

[8] A. Goudar and A. P. Schoellig, "Online spatio-temporal calibration of tightly-coupled ultrawideband-aided inertial localization," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1161–1168.

[9] Z. Cao, R. Liu, C. Yuen, A. Athukorala, B. K. K. Ng, M. Mathanraj, and U.-X. Tan, "Relative localization of mobile robots with multiple ultra-wideband ranging measurements," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5857–5863.

[10] Y. Song, M. Guan, W. P. Tay, C. L. Law, and C. Wen, "Uwb/lidar fusion for cooperative range-only slam," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6568–6574.

[11] M. J. Kuhn, J. Turnmire, M. R. Mahfouz, and A. E. Fathy, "Adaptive leading-edge detection in uwb indoor localization," in *2010 IEEE Radio and Wireless Symposium (RWS)*. IEEE, 2010, pp. 268–271.

[12] K. Bregar and M. Mohorčič, "Improving indoor localization using convolutional neural networks on computationally restricted devices," *IEEE Access*, vol. 6, pp. 17 429–17 441, 2018.

[13] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter, "Edge inference for uwb ranging error correction using autoencoders," *IEEE Access*, vol. 8, pp. 139 143–139 155, 2020.

[14] M. Ridolfi, J. Fontaine, B. Van Herbruggen, W. Joseph, J. Hoebeke, and E. De Poorter, "Uwb anchor nodes self-calibration in nlos conditions: a machine learning and adaptive phy error correction approach," *Wireless Networks*, vol. 27, no. 4, pp. 3007–3023, 2021.

[15] M. Zhao, T. Chang, A. Arun, R. Ayyalasomayajula, C. Zhang, and D. Bharadia, "Uloc: Low-power, scalable and cm-accurate uwb-tag localization and tracking for indoor applications," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–31, 2021.

[16] C. Mao, K. Lin, T. Yu, and Y. Shen, "A probabilistic learning approach to uwb ranging error mitigation," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[17] L. Yang and G. B. Giannakis, "Ultra-wideband communications: an idea whose time has come," *IEEE signal processing magazine*, vol. 21, no. 6, pp. 26–54, 2004.

[18] "Dw1000 datasheet," https://www.decawave.com/sites/default/files/resources/dw1000-datasheet-v2.09.pdf, accessed: 2021-12-10.

[19] "Decawave trek1000: Two-way-ranging (twr) rtls ic evaluation kit," https://www.decawave.com/wp-content/uploads/2018/09/trek1000_product_brief1.3.pdf, accessed: 2021-12-10.

[20] D. Neirynck, E. Luk, and M. McLaughlin, "An alternative double-sided two-way ranging method," in *2016 13th workshop on positioning, navigation and communications (WPNC)*. IEEE, 2016, pp. 1–4.

[21] S. Marano, W. M. Gifford, H. Wymeersch, and M. Z. Win, "Nlos identification and mitigation for localization based on uwb experimental data," *IEEE Journal on selected areas in communications*, vol. 28, no. 7, pp. 1026–1035, 2010.

[22] J. Khodjaev, Y. Park, and A. S. Malik, "Survey of nlos identification and error mitigation problems in uwb-based positioning algorithms for dense environments," *annals of telecommunications-annales des télécommunications*, vol. 65, no. 5, pp. 301–311, 2010.

[23] V. Barral, C. J. Escudero, J. A. García-Naya, and R. Maneiro-Catoira, "Nlos identification and mitigation using low-cost uwb devices," *Sensors*, vol. 19, no. 16, p. 3464, 2019.

[24] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo, "Uwb nlos/los classification using deep learning method," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2226–2230, 2020.

[25] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-line-of-sight identification and mitigation using received signal strength," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 1–1, 01 2014.

[26] Y.-S. Hsiao, M. Yang, and H.-S. Kim, "Super-resolution time-of-arrival estimation using neural networks," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1692–1696.

[27] U. Raza, A. Khan, R. Kou, T. Farnham, T. Premalal, A. Stanoev, and W. Thompson, "Dataset: Indoor localization with narrow-band, ultra-wideband, and motion capture systems," 11 2019, pp. 34–36.

[28] S. Kram, M. Stahlke, T. Feigl, J. Seitz, and J. Thielecke, "Uwb channel impulse responses for positioning in complex environments: A detailed feature analysis," *Sensors*, vol. 19, no. 24, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/24/5547

[29] "Vicon products," https://www.vicon.com, accessed: 2021-12-10.

[30] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. de Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham, "milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 109–122.

[31] "Ultra puck," https://velodynelidar.com/products/ultra-puck/, accessed: 2021-12-10.

[32] J. Karedal, S. Wyne, P. Almers, F. Tufvesson, and A. F. Molisch, "Statistical analysis of the uwb channel in an industrial environment," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, vol. 1. IEEE, 2004, pp. 81–85.