POLITECNICO DI TORINO
Repository ISTITUZIONALE

Ultra-Low-Power Range Error Mitigation for Ultra-Wideband Precise Localization

Original
Ultra-Low-Power Range Error Mitigation for Ultra-Wideband Precise Localization / Angarano, S., Salvetti, F., Mazzia, V., Fantin, G., Gandini, D., Chiaberge, M.. - ELETTRONICO. - 507:(2022), pp. 814-824. (Proceedings of the 2022 Computing Conference London (UK) July 14-15, 2022) [10.1007/978-3-031-10464-0_56].

Availability:
This version is available at: 11583/2970110 since: 2022-07-14T10:43:30Z

Publisher:
Springer

Published
DOI:10.1007/978-3-031-10464-0_56

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-10464-0_56

(Article begins on next page)

12 October 2026

Ultra-low-power Range Error Mitigation for Ultra-wideband Precise Localization

Simone Angarano[1,2], Francesco Salvetti[1,2,3] Vittorio Mazzia[1,2,3], Giovanni Fantin[1,2], Dario Gandini[1,2], and Marcello Chiaberge[1,2]

[1] Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy
[2] PIC4SeR, Politecnico di Torino Interdepartmental Centre for Service Robotics, Turin, Italy
[3] SmartData@PoliTo, Big Data and Data Science Laboratory, Turin, Italy
{name.surname}@polito.it

Abstract. Precise and accurate localization in outdoor and indoor environments is a challenging problem that currently constitutes a significant limitation for several practical applications. Ultra-wideband (UWB) localization technology represents a valuable low-cost solution to the problem. However, non-line-of-sight (NLOS) conditions and complexity of the specific radio environment can easily introduce a positive bias in the ranging measurement, resulting in highly inaccurate and unsatisfactory position estimation. In the light of this, we leverage the latest advancement in deep neural network optimization techniques and their implementation on ultra-low-power microcontrollers to introduce an effective range error mitigation solution that provides corrections in either NLOS or LOS conditions with a few mW of power. Our extensive experimentation endorses the advantages and improvements of our low-cost and power-efficient methodology.

Keywords: Deep Learning, Edge AI, Ultra-Wideband, Indoor Positioning

1 Introduction

As Global Navigation Satellite System (GNSS) is the benchmark solution for outdoor positioning, Ultra-wideband (UWB) real-time locating systems (RTLS) have recently become the state of the art technology for localization in indoor environments [13]. Indeed, with its high signal frequency and very narrow pulses, UWB outperforms all other wireless positioning systems like WiFi and BLE thanks to its decimeter level of precision and higher resilience to multipath effects [9].

Nevertheless, in a real-world scenario, the complexity of the environment often leads to partial or total obstruction of the signal between the transmitter and the receiver, thus causing a substantial degradation of the positioning performances. The non-line-of-sight (NLOS) condition affects the time-of-arrival

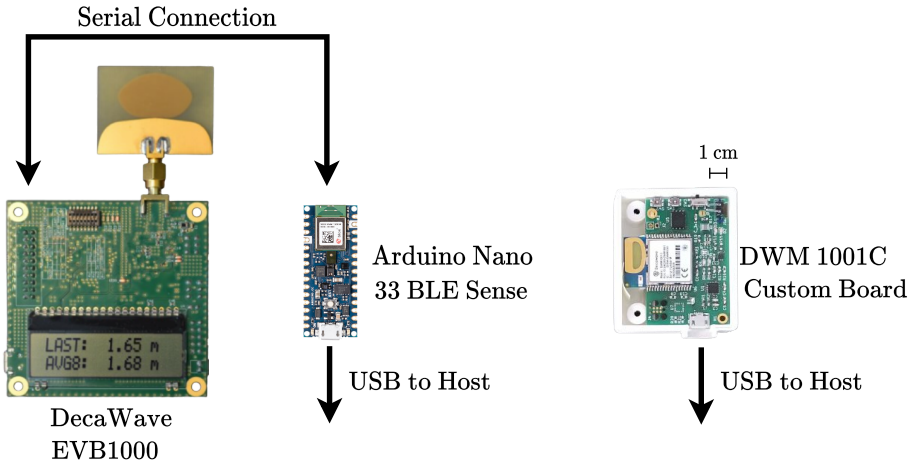

Fig. 1. Hardware setup for ultra-low-power UWB range error mitigation. The DecaWave EVB1000 board is connected to a power supply and an external microprocessor (Arduino Nano 33 BLE Sense) that locally runs a highly optimized and power-efficient deep neural network for range error mitigation. We also show our custom board designed for the DWM1001C module for size comparison. Future works will fully integrate our methodology on our custom board, providing a compact solution for precise localization.

(ToA) measurement introducing a positive bias in the ranging estimation [8]. Moreover, multipath components also strongly influence range estimates, especially in indoor environments where walls and furniture are often made of reflecting materials.

Therefore, to achieve better localization accuracy, a mitigation algorithm is needed, and this must be robust and general enough to be effective in a large set of different scenarios. Most NLOS identification and mitigation methodologies proposed in the literature are based on channel impulse response (CIR) statistics [2], likelihood ratio or binary hypothesis tests [10], and machine learning techniques. As regards the latter, several techniques have been investigated, such as representation learning models [12], support vector machines (SVM), [16] and Gaussian processes (GP) [15]. Despite the chosen methodology, the resulting mitigation algorithm must require a low computational effort to be usable in a real-world use case and consequently come out of a pure research scenario. Indeed, most applications, like robotic indoor navigation and person or object tracking, typically make use of single-board computers with limited computational capabilities and stringent power consumption requirements.

In this research project, we propose a highly optimized deep learning model for range error mitigation that requires a few mW to compensate NLOS and LOS signals. The proposed methodology can run at high frequency on ultra-low-power microcontrollers, enabling the design of small and low-power devices

for precise indoor localization. The complete setup used in our experimentation is presented in Figure 1. The main contributions of this paper are the following.

- Introduce UWB range error mitigation for ultra-low-power microcontrollers with deep learning at the edge.
- Modify and highly optimize a deep learning model, leveraging the latest weight quantization and graph optimization techniques for power and latency reduction.
- Evaluate the real-time performance of the resulting network, measuring latency, power, and energy usage with different CIR sizes.

The rest of the paper is organized as follows. Section 2 presents the proposed methodology with a detailed explanation of the network and the adopted optimization techniques for power consumption and latency reduction. Section 3 presents the experimental results and discussion after briefly describing the DeepUWB dataset used for the tests. Finally, section 4 summarizes the main achievements of the work and proposes future research developments.

2 Methodology

In this section, we present the proposed methodology for the embedded implementation of a UWB error mitigation algorithm on an ultra-low-power microcontroller. We present details on the deep neural network design and the techniques adopted to optimize it, quantize all its parameters to 8 bits integers, and deploy the final model on the target board.

We model the mitigation process as presented by Angarano et al. [1]:

$$\hat{d} = d + \Delta d \tag{1}$$

where the goal is to predict an estimate of the error Δd on the UWB range measurement in order to compensate the observed quantity $\hat{d}$ and recover the true distance d. Therefore, we adopt a DNN model that predicts an estimate $\hat{y}$ of the true latent error $y = \Delta d$ as a non-linear function of the input CIR vector X, measured by the UWB sensor.

We denote with K the number of temporal samples of the CIR. We develop our methodology so that the dimension K can be changed to analyze its effects on the computational efficiency and the accuracy of the algorithm.

2.1 Network Design

The original design of the Range Error Mitigation Network (REMNet) presented in [1] is adapted to be executed in real-time on a low-power microcontroller. The neural network should be fully quantized to perform all the operations with 8-bit integers and meet the real-time constraints. So, in order to overcome software and hardware limitations of standard low-power microcontroller solutions, we modify

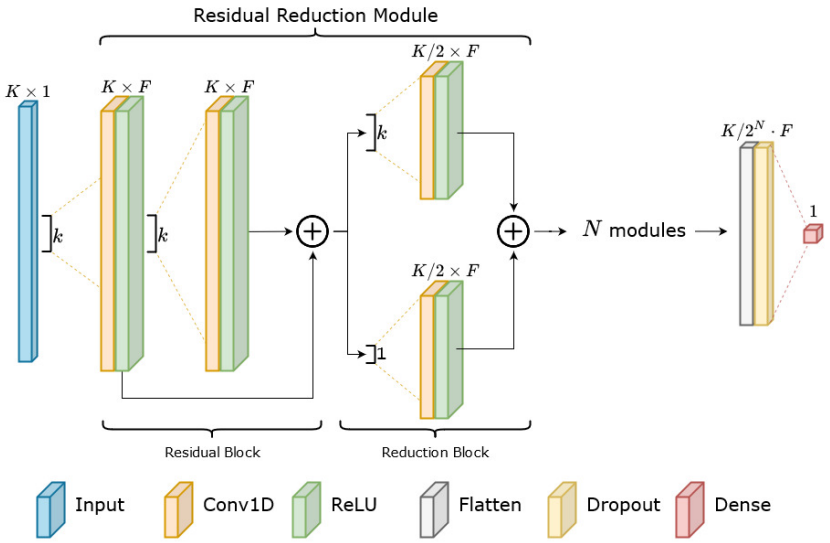

Fig. 2. Range Error Mitigation Network (REMNet) architecture modified to ensure compatibility with the target embedded board. The input of the model is the $K \times 1$ tensor representing the CIR of the measurement. Subsequent N Residual Reduction Modules progressively reduce the original dimension K. Finally, a fully connected layer composes the high-level extracted features of dimension $K/2^N \cdot F$ and outputs the range error estimation.

the original REMNet architecture, removing all self-attention blocks that boost the accuracy performance but compromise compatibility.

The overall modified architecture of the REMNet model is shown in Figure 2. Starting from the input tensor CIR $\boldsymbol{X}$ of size $K \times 1$, we extract low-level features with a first 1D convolution operation with a kernel of dimension k_0. The core of REMNet is the residual reduction module (RRM). Firstly, the residual is computed with respect to a 1D convolution of kernel k_n; then, a reduction block decreases the temporal dimension K with a 1D convolution with a stride of 2. The reduction block again has a residual connection characterized by a 1D convolution with a kernel k of dimension 1 and stride 2 to match the temporal dimension.

Overall, each RRM block computes the following non-linear mapping function:

$$\mathrm{RRM}(\boldsymbol{X}) = \mathrm{Red}(\mathrm{Conv1D}(\boldsymbol{X}) + \boldsymbol{X}) \tag{2}$$

where

$$\mathrm{Red}(\boldsymbol{X}) = \mathrm{Conv1D}_{s=2}(\boldsymbol{X}) + \mathrm{Conv1D}_{k=1;s=2}(\boldsymbol{X}) \tag{3}$$

The network is characterized by a stack of N RRM, all with ReLU as non-linear activation functions [7]. After N RRM blocks, we obtain a tensor with shape $K/2^N \times F$. We perform a flattening operation to feed a regression head

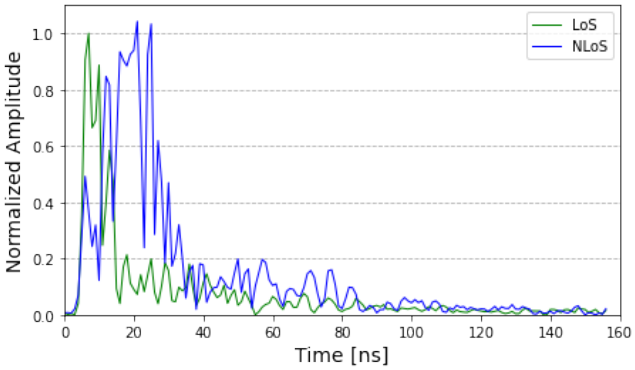

Fig. 3. LoS and NLoS samples from the DeepUWB dataset, with normalized amplitude. In the NLoS case, the signal travels along many routes until it reaches the antenna. That makes the ToA estimation ambiguous, introducing a positive bias in the ranging measurement.

composed of dropout and a fully connected layer that predicts the final estimate of the compensation value Δd. We denote with F the number of features of each convolutional operation. We always use zero padding and the same value k_n for each convolutional kernel.

2.2 Network Optimization and Quantization Techniques

To achieve the goal of a real-time implementation, the range error mitigation technique must respect constraints on memory, power, and onboard latency. We study different graph optimization and quantization methods to reduce computational cost without compromising performance. Several techniques have been developed to increase model efficiency in the past few years [4], from which the following methods are chosen. First, network pruning and layer fusing are applied to remove nodes and operations that give almost no contribution to the output. Moreover, the number of bits used to represent network parameters and activation functions is reduced by quantizing the float32 values to int8 ones. Combining these strategies strongly increases efficiency with minimal impact on performance.

Graph optimization is first applied to the model trained in plain float32 without quantization to investigate its effects on accuracy and dimension. Finally, a third version of the network is obtained by quantizing weights, activations, and math operations through scale and zero-point parameters. We follow the methodology presented by Jacob et al. [4], in which each weight and activation are quantized with the following equation:

$$r = S(q - Z) \tag{4}$$

where r is the original floating-point value, q the quantized integer value, and S and Q are the quantization parameters (respectively scale and zero point). A

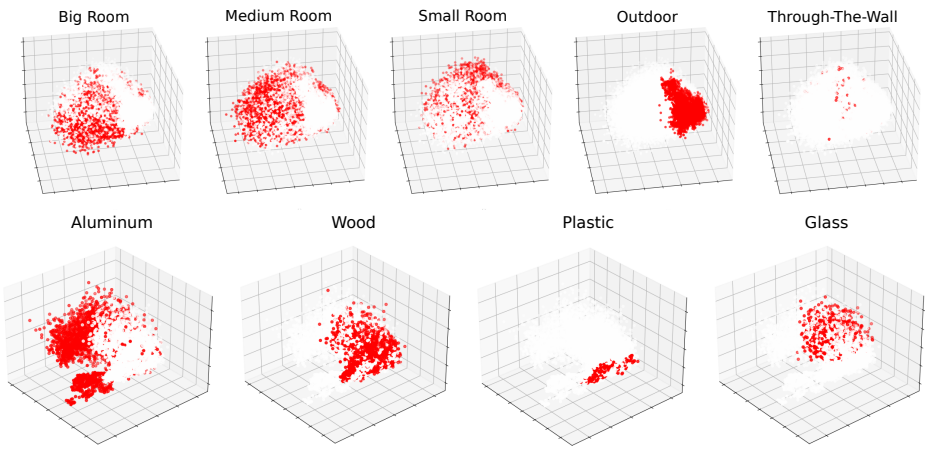

Fig. 4. Principal Component Analysis representation of DeepUWB benchmark dataset, showing different spatial configurations for different rooms and materials. The original CIR dimensions are projected into a three-dimensional space. The first row shows the data point projection divided into the five considered environments. On the other hand, the second row highlights the effects of materials on signal propagation. It is clear how different molecular structures affect the signal in different ways.

fixed-point multiplication approach is adopted to cope with the non-integer scale of S. This strategy drastically reduces memory and computational demands due to the high efficiency of integer computations on microcontrollers. The final step is to convert and import the quantized model into the embedded application system. As most microcontrollers do not have the resources to run a filesystem, we provide the network in a C source file that can be included in the program binary and loaded directly into memory, as suggested by [14]. All the results obtained with the models at different quantization steps are presented in Section 3.

3 Experiments and Results

In this section, we perform an experimental evaluation of different optimized versions of REMNet. Moreover, we test the accuracy and performance of the network on a low-cost microcontroller-based development board, reporting inference speed, and power consumption.

3.1 The DeepUWB Dataset

In the following experiments, we employ the indoor samples of the DeepUWB dataset presented in [1] and publicly available on Zenodo[4]. The data is obtained

[4] http://doi.org/10.5281/zenodo.4290069

using DecaWave EVB1000 transmitters and taking several LOS and NLOS in different indoor and outdoor environments in the presence of various types of obstacles. Figure 3 presents a comparison between LoS and NLoS samples from the dataset. Range estimates taken in NLoS conditions are typically positively biased [8].

For each of the 55,000 measures, both ground-truth distance and the one given by the UWB boards are included, as well as the environment scenario, the obstacle materials, and the CIR vector used as input for REMNet. Three differently sized rooms are selected for indoor measurements to cover various office-like situations: a large one (10m x 5m), a medium one (5m x 5m), and a small one (5m x 3.5m). Regarding obstacles, various typical objects for an indoor scenario are used to cover a wide range of materials, including plastic, glass, metal, and wood.

Figure 4 shows the result of Principal Component Analysis (PCA) on DeepUWB: it is noticeable that the three indoor scenarios occupy close areas in the 3D space, very distinct from outdoor and through-the-wall measurements. That is due to the presence of strong multipath components. Measurements taken in the presence of different materials tend to occupy different regions, with heavier and more screening ones, such as aluminum, being highly concentrated and distant from materials like plastic and wood.

3.2 Experimental Setting

We keep aside the medium-sized room measurements as our primary scope is to evaluate the effectiveness of REMNet in compensating the error for general indoor scenarios. In total, 36023 and 13210 training and testing data points are used, respectively. LoS and NLoS samples are kept together in the sets to evaluate the performance of the network in both cases. Real-time range mitigation with the whole CIR vector could be very computationally intensive [17]. For this reason, a study is conducted on the number of samples necessary to have an acceptable error correction. Then, the Tensorflow Lite [5] framework is used to perform graph optimization and to quantize weights, activations, and math operations. The final test measures the inference frequency of the model deployed on an Arduino Nano 33 BLE Sense[6], alongside its power usage.

In order to select the optimal number of input features, we conduct a grid search study on the number of CIR temporal samples. We progressively reduce the input dimension K from 157, suggested in [3], to 8. The Mean Absolute Error (MAE) is used as the loss function and metric. Box plot of model performance with the different CIR input sizes is shown in Figure 5.

The network hyperparameters are obtained with an initial random search followed by a grid search exploration to fine-tune them and compromise accuracy and efficiency. We use $N = 3$ residual reduction modules with kernel dimensions $k_0 = 5$ and $k_n = 3$, and $F = 16$ filters.

[5] https://www.tensorflow.org/lite
[6] https://store.arduino.cc/arduino-nano-33-ble-sense

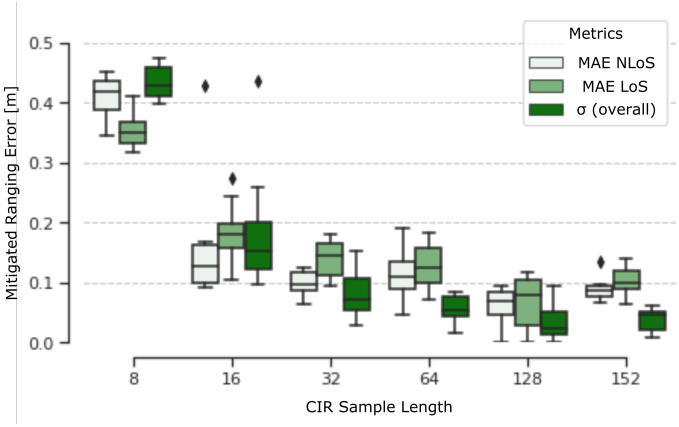

Fig. 5. REMNet performance (mitigated ranging error) with different CIR input size dimensions. For each test, we report LoS and NLoS MAE as well as overall standard deviation (σ). It is clear how a reduced number of input features degrades the performance of the model. Moreover, an input with eight dimensions appears to be the minimum amount of information required to obtain an acceptable range error estimation.

Table 1. Model performance for different CIR lengths before and after applying optimizations. The results for a Multilayer Perceptron are included as a reference.

Model	CIR	Params	MAE [m]	MAE_{GO} [m]	MAE_{INT8} [m]
REMNet	157	5905	0.0687	0.0687	0.0690
	128	5841	0.0702	0.0702	0.0698
	64	5713	0.0704	0.0704	0.0701
	32	5649	0.0710	0.0710	0.0713
	16	5617	0.0712	0.0712	0.0714
MLP	157	54401	0.0769	0.0777	0.0775

Finally, all experimentation adopt Adam as the optimization algorithm [6] with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The optimal learning rate $\lambda = 3e - 4$ is experimentally derived using the methodology described in [11] and kept constant for 30 epochs, with a batch size of 32. We employ the TensorFlow framework [7] to train the network on a PC with 32 GB RAM and an Nvidia 2080 Super GP-GPU. The overall training process can be performed in less than 5 minutes.

3.3 Quantitative Results

The medium room data samples, used as test set, have a starting MAE of 0.1242 m and a standard deviation of $\sigma = 0.1642$ m. The results obtained by the trained

[7] https://www.tensorflow.org/

Table 2. Model dimensions for different CIR lengths before and after applying optimizations. The results for a Multilayer Perceptron (MLP) are included as a reference.

Model	CIR	Params	Dim [kB]	Dim_{GO} [kB]	Dim_{INT8} [kB]
REMNet	157	5905	317.321	32.988	23.088
	128	5841	317.321	32.732	23.024
	64	5713	317.157	32.220	22.896
	32	5649	317.052	31.964	22.832
	16	5617	317.499	31.836	22.800
MLP	157	54401	216.047	125.885	60.320

Table 3. Real-time performance for different optimized models, including inference frequency, consumed power, and network energy usage.

Model	CIR	f_m [Hz]	V_{cc} [V]	I_{abs} [mA]	P_{abs} [mW]	E_{inf} [mJ]
REMNet	157	17.2	3.3	16.2	53.4	3.1
	128	21.2	3.3	16.0	52.8	2.5
	64	41.0	3.3	15.8	52.2	1.3
	32	77.8	3.3	15.6	51.6	0.66
	16	140.0	3.3	15.6	51.6	0.37
MLP	157	184.1	3.3	16.2	53.4	0.29

reference architectures and their degradation, as optimizations are applied, are shown in Table 1. Each model has been tested five times with different random seeds to obtain statistically significant results. Performances prove the effectiveness of the model, as the MAE of the medium room samples is reduced by 45.7% using the reference model. Consequently, the final error of 0.0687 m is comparable to the actual LoS precision of EVB1000 boards [5]. Lastly, REMNet demonstrates to outperform a Multilayer Perceptron (MLP) with around 10% of its parameters.

As regards model optimization, columns *CIR* and *Params* report the number of input samples used for the mitigation and the total model parameters, respectively. Moreover, the resulting MAE is reported for all three model configurations: reference, graph optimization (MAE_{GO}), and full 8-bit integer quantization (MAE_{INT8}). The results show that the effect of graph optimization is null for REMNet, while the MLP performance slightly deteriorates. Integer quantization, instead, minimally increases the resulting MAE for all the models. Finally, our experimentation confirms that fewer dimensions of 128 tend almost linearly to degrade the network's accuracy.

Despite the insignificant effect of graph optimization on performance, memory occupancy greatly shrinks. Our results show a reduction of about 90% for REMNet, while the MLP only halves its memory footprint due to its higher number of parameters. In addition, quantization allows a further reduction of REMNet memory requirements of an additional 30%, confirming the great benefit of using both optimization and quantization techniques provided by TensorFlow Lite converter. The MLP reduces another 50% of its memory footprint

with full integer quantization, reaching a final size of about three times REMNet. Therefore, the proposed model can outperform the baseline both in error mitigation capability and memory requirements. All the results on the memory footprint of the models under examination are presented in Table 2.

Finally, the inference speed and the power consumed by the Arduino board for each considered model are analyzed and presented in Table 3. The frequencies, denoted as f_m, have been measured as the reciprocal of the maximum inference time over a series of tests. In all the cases, they can be considered compliant for real-time applications. In particular, the MLP requires less computational effort despite the more significant number of parameters because it involves simpler math operations than REMNet. Moreover, reducing CIR length results in an almost linear increase in inference speed. To assess power consumption, instead, we measured the absorbed current I_{abs} with a voltage supply $V_{cc} = 3.3$ V. Results show that, as the microcontroller is constantly processing data, the power usage can be considered constant for all the cases. However, since inference speed significantly changes with model complexity, we computed the energy required for a single inference step E_{inf} by dividing the consumed power P_{abs} by the frequency f_m. It is noticeable that very few mJ are sufficient to run range mitigation, reaching values under 1 mJ. That proves that the already efficient design of the proposed model, in conjunction with 8-bit weight precision and graph optimization techniques, makes deep learning a feasible solution for effective ultra-low-power UWB range error mitigation.

4 Conclusions

In this paper, we introduced UWB range error mitigation for ultra-low-power microcontrollers. Our effective and power-efficient methodology builds on top of the latest advancement in deep learning and neural networks optimization techniques to provide precise localization in NLOS and LOS conditions with a few mW of power. We proposed a modified version of REMNet, a lightweight model designed explicitly for range error mitigation on ultra-low-power AI edge devices. Our extensive experimentation proves how the proposed system successfully runs at a high frequency on microcontrollers and provides enhanced localization in indoor environments. Future works will integrate the proposed methodology on a compact custom board designed around the DWM1001C DecaWave module to provide a compact solution for precise localization.

Acknowledgments

This work is partially supported by the Italian government via the NG-UWB project (MIUR PRIN 2017) and developed with the contribution of the Politecnico di Torino Interdepartmental Center for Service Robotics PIC4SeR[8] and SmartData@Polito[9].

[8] https://pic4ser.polito.it
[9] https://smartdata.polito.it

References

1. Angarano, S., Mazzia, V., Salvetti, F., Fantin, G., Chiaberge, M.: Robust ultra-wideband range error mitigation with deep learning at the edge. Engineering Applications of Artificial Intelligence **102**, 104,278 (2021)
2. Barral, V., Escudero, C.J., García-Naya, J.A.: Nlos classification based on rss and ranging statistics obtained from low-cost uwb devices. In: 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5. IEEE (2019)
3. Bregar, K., Mohorčič, M.: Improving indoor localization using convolutional neural networks on computationally restricted devices. IEEE Access **6**, 17,429–17,441 (2018)
4. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2704–2713 (2018)
5. Jiménez, A.R., Seco, F.: Comparing decawave and bespoon uwb location systems: Indoor/outdoor performance analysis. In: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1–8. IEEE (2016)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Icml (2010)
8. Otim, T., Díez, L.E., Bahillo, A., Lopez-Iturri, P., Falcone, F.: Effects of the body wearable sensor position on the uwb localization accuracy. Electronics **8**(11), 1351 (2019)
9. Schmid, L., Salido-Monzú, D., Wieser, A.: Accuracy assessment and learned error mitigation of uwb tof ranging. In: 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1–8. IEEE (2019)
10. Silva, B., Hancke, G.P.: Ir-uwb-based non-line-of-sight identification in harsh environments: Principles and challenges. IEEE Transactions on Industrial Informatics **12**(3), 1188–1195 (2016)
11. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472. IEEE (2017)
12. Stahlke, M., Kram, S., Mutschler, C., Mahr, T.: Nlos detection using uwb channel impulse responses and convolutional neural networks. In: 2020 International Conference on Localization and GNSS (ICL-GNSS), pp. 1–6. IEEE (2020)
13. Tiwari, P., Malik, P.K.: Design of uwb antenna for the 5g mobile communication applications: A review. In: 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM), pp. 24–30. IEEE (2020)
14. Warden, P., Situnayake, D.: TinyML. O'Reilly Media Inc. (2019)
15. Xiao, Z., Wen, H., Markham, A., Trigoni, N., Blunsom, P., Frolik, J.: Non-line-of-sight identification and mitigation using received signal strength. IEEE Transactions on Wireless Communications **14**(3), 1689–1702 (2014)
16. Ying, R., Jiang, T., Xing, Z.: Classification of transmission environment in uwb communication using a support vector machine. In: 2012 IEEE Globecom Workshops, pp. 1389–1393. IEEE (2012)
17. Zeng, Z., Liu, S., Wang, L.: Nlos identification for uwb based on channel impulse response. In: 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1–6. IEEE (2018)