Track 7: Channel Impulse Responses

8th IPIN Competition off-site Indoor Localization, version 3.0

Sebastian Kram[1,2] Maximilian Stahlke[1] Christopher Mutschler[1]
{firstname.lastname}@iis.fraunhofer.de

[1]Fraunhofer Institute for Integrated Circuits IIS
[2]Friedrich-Alexander-University Erlangen-Nürnberg (FAU)

1 Introduction and Scope

Radio-Frequency (RF) positioning in cluttered indoor environments is challenging. As signals travel through the environment along different paths it is difficult to determine the correct time-of-flight (TOF) of the transmitted signals. Traditionally, fingerprinting-based solutions have been used to estimate a rough position from narrow-band signals such as Wi-Fi or Bluetooth. However, with modern ultra-wideband (UWB) technology signals can be transmitted at higher bandwidths, enabling a much higher spatial resolution from which we can the extract complex propagation conditions such as absorption, reflection, diffraction and scattering [1]. While UWB is not yet integrated in consumer devices, current progress in development and standardization make it likely that they will be ubiquitous in the near future. This allows for low-cost ad-hoc positioning.

To leverage the benefits of the high spatial solution we can make use of the *channel information* (CI). For sufficiently high bandwidths the CI roughly corresponds to the complex-valued *channel impulse response* (CIR). Recently, these signals have been used for positioning in different ways [2]:

- *Model Error Mitigation*: The CI is used to classify propagation conditions like non-line-of-sight (NLOS) or to estimate time-of-flight errors caused by obstructed LOS (OLOS). This enhances classic tracking algorithms by providing additional information on the channel states [3].

- *Fingerprinting*: The propagation conditions are assumed to cause significant differences in the spatial behavior of the CI, which can be exploited by comparing them with previously recorded data (either using the CI or extracted features). For Machine Learning (ML) and Deep Learning (DL) approaches this constitutes a regression task [4].

- *Multipath-SLAM* : The CI (or extracted multipath components, MPCs) are used to jointly estimate virtual anchors (i.e. characteristic reflection points caused by reflecting surfaces) and the trajectory of the transmitter. The main challenge is to correctly associate the extracted MPCs with specific surfaces or reflecting objects [5].

Apart from these main concepts, various different or hybrid approaches exist, each of them with its distinct advantages or disadvantages. We present a dataset that contains a realistic indoor tracking scenario in an industrial setting to allow for a fair comparison for practical application.

2 Environment and Measurement Setup

The environment consist of an area of approx $300m^2$ with a partial enclosure by reflecting walls (consisting of the walls of the measurement wall, including metal gates and artifically included reflector/absorber walls elements) and various metal objects that are typical for industrial indoor environments, like e.g. industrial vehicles or metal shelves. Fig. 1 schematically sketches the environment, while Fig. 2 shows a picture of the actual setup: receiving anchors are placed around the recording area at $\sim 1.5m$ height. The transmitter device is carried by a human/worker and regularly transmits UWB signals received by the anchors. The data is recorded using a platform based on the Decawave DW1000 UWB chip.

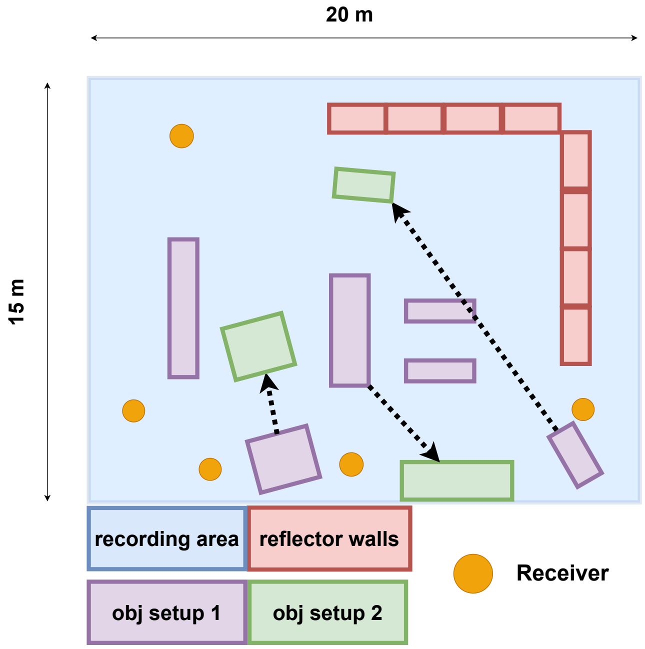

Figure 1: Schematic environment setup, including exemplary object setups 1 and 2.

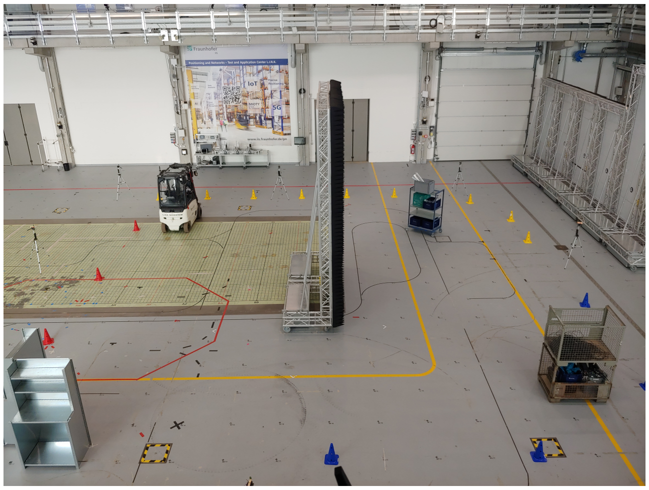

Figure 2: Image of an environment similar to the one used for the challenge.

The ground truth of the transmitter position is collected using a millimeter-accurate Qualisys motion tracking system. The data is collected and synchronized by an NTP server and pre-processed (corrupted datapoints are removed and RF and positioning reference data are synchronized).
This challenge contains two scenarios:

1. For the first scenario we provide training data with ground truth positional information and the models submitted by the competitors will be evaluated on a test set (a few trajectories) that originate from the same measurement campaign, i.e., training and test data have been recorded on the same environmental setup. Both training and test data contain complete trajectories while the trajectories of the test data set are shorter. The test set does not contain ground truth position labels.

2. The second scenario presents a modification of the first scenario. In this setup we moved clutter elements within the environment (e.g. forklift, van, etc.) which lead to a slightly different propagation scenario. We will not provide training data for the modified version of the scenario but only test data (without ground truth). The goal of this scenario is to test if the models submitted by the competitors over-fit to the previous environmental setup and fail to generalize well to changes to the environment.

3 Dataset description

3.1 Main Dataset

The main dataset is provided as a HDF5 file, that can be loaded by various environments. The main file (i.e. the training data) contains the CI and reference positions. Each data instance (i.e. column of the dataframe) contains:

- rec_time ([float]): the timestamp in s at which the CIR was received at the receiver node. (This is the ”global time index” of the tracking problem)
- ci_time (array[float]): the timestamps corresponding to the imaginary and real parts of the CI in s. (This is the ”local time index” that can be used to assign a distance to the CI values)
- burst_id ([int]): the transmitter time index. This can be used for synchronization. For clarity, at each of the burst IDs, the transmitter (i.e., the mobile node) transmits an impulse that is received by a subset of the receivers (i.e, anchors). The complete set of CIRs from all anchors is not available at all time steps (as at some receivers the detection was not successful due to an insufficient channel and/or data corruption).
- ci_real (array[int]) and ci_imag (array[int]): the real and imaginary parts of the CI as tuples. The CI is centered around the first distinct peak and contains 366 samples each, which can be set in relation to distance or time-of-flight by using cir_time, as depicted in Fig. 3.
- anch_id ([string]): the anchor id of the receiving anchor.
- The positions of the agent (i.e. the mobile tag, the transmitter) ref_x, ref_y as float. The reference positions are corresponding to the receiver timestamp rec_time.

An additional .TXT-file (anchors.txt) containing the anchor/receiver positions is also available. It contains:

- anch_ID [string] the anchor IDs.
- p_x, p_y [float] positions of the anchors.

An additional .TXT-file (hall$_measurements.txt)containingthecornercoordinatesoftheabsorber/reflectorw$

x1,y1,x2,y2 [float] The corner coordinates of the walls.

description [float] A short description of the type of wall.

Note: *We recommend using Python 3.7 or 3.8 and loading the data by using the pandas library https://pandas.pydata.org/docs/reference/api/pandas.read_hdf.html, if you have problems with the data format please do not hesitate to contact us.*

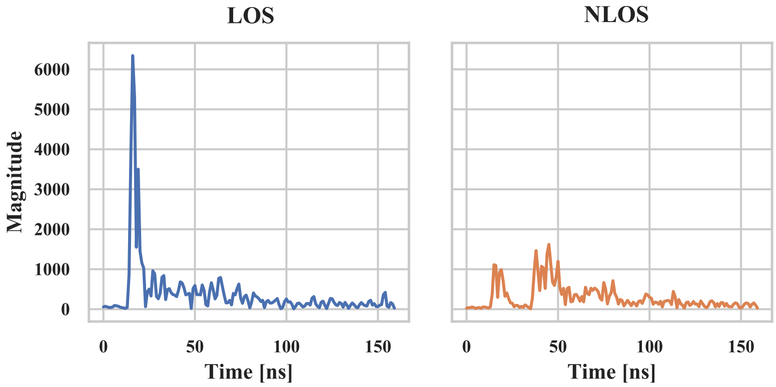

Figure 3: Visualization of two exemplary recordings in a LOS and NLOS case: the time labels of the x-axis are given in `ci_time`, the corresponding magnitudes on the y-axis are given by the complex numbered array defined by `ci_real` and `ci_imag`

.

4 Challenge Objectives

The challenge is divided into two parts. In the first part the data that is used for training and testing originate from the same environment setup. In the second part, we made some changes to the environment setup (i.e., we moved mobile metallic objects) in order to consider the robustness of the algorithms to environmental changes. For the second scenario we do not provide training data but only test data. The trajectories we use for testing in the second scenario stay within a similar area as the one used in the first scenario.

For both scenarios, we provide a data frame that includes the same columns as the training dataset (obviously, apart from the reference positions `ref_x`, `ref_y`). Additionally, for each setup the initial position, perturbed by artificial additive zero-mean white Gaussian noise of standard deviation $1m$ in x and y-directions, is available. For the first test it is $[27.19, 10.42]$m, for the second $[29.82, 22.74]$m. For each of the files, the competitors are to provide a file with results. The overall duration of both test datasets is 190 s. Using these data, the objective is to estimate position values at a regular time interval of 0.5s. The participants must provide two .HDF- or .CSV files of length 380, labeled `results1` and `results2` that include:

- the timestamps `t_est` [`float`] of the position estimates in s.
- the corresponding estimated positions `x_est` and `y_est` [`float`].

Results should be provided for times $[0.5, 1.0, ..., 189.0, 189.5, 190.0]$s.

UPLOAD: Please include the two files into a `.zip`-file with a name clearly indicating your team and upload the file to `https://owncloud.fraunhofer.de/index.php/s/eIrdyMTXvLnpfV6`. If a team submits multiple files, we will only evaluate the last file.

Note: *For both of these setups, predicted positions need to be provided by all participants. It is up to the participants if they provide different solutions for both scenarios.*

Note: *For clarity, the input data are not available at a perfectly regular sampling interval and the complexity of CIRs does not allow for direct interpolation to obtain regularly spaced data, so, depending on the applied methods, re-sampling of the resulting position estimates might be necessary.*

5 Exemplary approaches

The objective of the challenge is to use the presented sets of CI to estimate the position of the tracked object. As mentioned in the introduction, different categories of positioning algorithms are possible for this task. For clarification, we included a *highly simplified* description of a possible pipeline for each category.

Model Error Mitigation: An exemplary tracking pipeline could look like the one depicted in Fig. 4: A ToF Estimation (Peak Tracking) algorithm is used to identify the strongest peak in the CI implying the distance between transmitter and receiver. An error mitigation algorithm, e.g. a machine learning approach, trained on the available training data is also applied on the CI to estimate an estimation error describing the difference in estimated an geometric difference caused by environment interaction. The corrected distance estimates are then processed in a tracking filter, producing a positioning result.

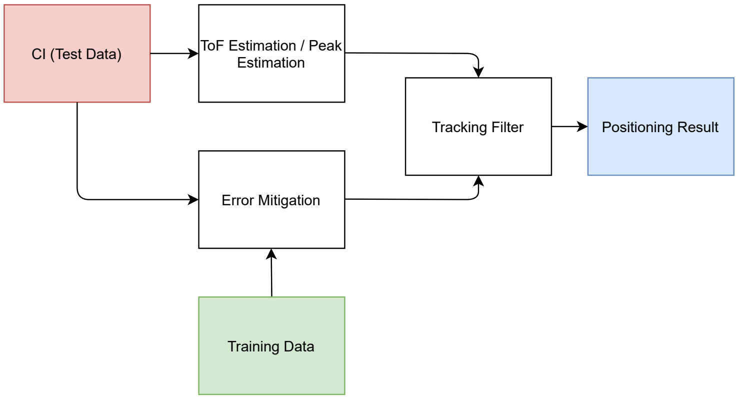

Figure 4: Exemplary Pipeline for a model error mitigation based system.

Fingerprinting: An exemplary positioning approach is sketched in Fig. 5: The positioning problem is seen as a regression task, where the input consists of the complete CI and the labels are the 2D-positions of the tracking target. For instance, a deep learning algorithm can be used for this regression task, producing positioning estimates, which are then smoothed using e.g. a Kalman filter.

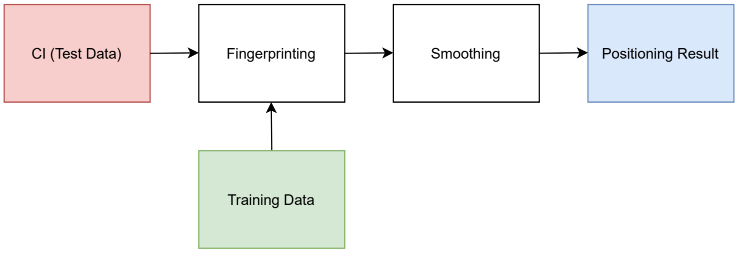

Figure 5: Exemplary Pipeline for a fingerprinting based system.

Channel SLAM: The presented dataset is not ideal for channel SLAM as it cannot directly benefit from the training data. To mitigate this, we included the coordinates of the reflector walls in the environment to initialize virtual anchor hypotheses. A typical pipeline for a channel SLAM is depicted in Fig. 6. Distinct multipath components (MPCs) are extracted from the CI using a channel estimation algorithm. The channel SLAM algorithm then processes these by data association with existing virtual anchors (i.e., characteristic reflecting surfaces) and new virtual anchor hypotheses and uses the associated spatial information for tracking e.g. in a Rao-Blackwellized particle filter.

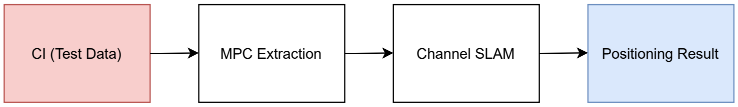

Figure 6: Exemplary Pipeline for a channel SLAM system.

6 Evaluation metrics

The Euclidean distance between estimated and true results (each 2D-positions) is the main evaluation metric. Specifically, third quartile is used as a performance metric. The results from the known and unknown environment test sets are weighted equally for performance.

7 Download

You can download the dataset at `https://owncloud.fraunhofer.de/index.php/s/RURgoPDou3PgF3U`. Please don't hesitate contact us for any questions you might have.

Upload: Please include the two files into a `.zip`-file with a name clearly indicating your team and upload the file to `https://owncloud.fraunhofer.de/index.php/s/eIrdyMTXvLnpfV6`. If a team submits multiple files, we will only evaluate the last file.

References

[1] A. Molisch, "Ultra-wide-band propagation channels," *Proceedings of the IEEE*, vol. 97, pp. 353 – 371, 03 2009.

[2] S. Aditya, A. F. Molisch, and H. M. Behairy, "A survey on the impact of multipath on wideband time-of-arrival based localization," *Proceedings of the IEEE*, vol. 106, no. 7, pp. 1183–1203, 2018.

[3] H. Wymeersch, S. Maranò, W. M. Gifford, and M. Z. Win, "A machine learning approach to ranging error mitigation for uwb localization," *IEEE transactions on communications*, vol. 60, no. 6, pp. 1719–1728, 2012.

[4] A. Niitsoo, T. Edelhäußer, and C. Mutschler, "Convolutional neural networks for position estimation in tdoa-based locating systems," in *Proc. 9th Intl. Conf. Indoor Positioning and Indoor Navigation, Nantes, France*, pp. 1–8, 2018.

[5] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann, and U.-C. Fiebig, "Multipath assisted positioning with simultaneous localization and mapping," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 1–1, 09 2016.