Track 7: Channel Impulse Responses

9th IPIN Competition off-site Indoor Localization, version 1.0

Sebastian Kram[1] Christian Nickel[1] Christopher Mutschler[1]
{firstname.lastname}@iis.fraunhofer.de

[1]Fraunhofer Institute for Integrated Circuits IIS

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

Figure 1: Image of the environment. The mobile robot can be seen on the right.

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.

While the focus of last year's challenge was the adaption to changed environments, this year it is the generalization to another agent: For the training and evaluation datasets, different agents, i.e., a mobile robot and a worker, are tracked.

2 Environment and Measurement Setup

The environment consist of a warehouse area of approx $1,200m^2$ with an enclosure by reflecting walls (consisting of the walls of the warehouse, including metal gates.) The environment contains various metal objects, like e.g. industrial vehicles or metal shelves. Fig. 1 shows a picture of a part of the warehouse. Receiving anchors are placed around the recording area at $\sim 1.5m$ height. The transmitter device is carried by the mobile agent / tracking target and regularly transmits UWB signals received by the anchors. For the data collection phase, it is attached to a mobile robot. For the evaluation phase, it is carried as a handheld by a human/worker. We provide an exemplary and representative evaluation/experiment dataset for adjusting models. The data is recorded using a platform based on the Decawave DW1000 UWB chip at a bandwidth of $499.2\mathrm{MHz}$ and center frequencies of $4-6$ GHz.

The ground truth of the transmitter position is collected using a millimeter-accurate 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).
The main challenge this year is *agent generalization*. The majority of the provided data for the validation and training are collected by a mobile robot, while the evaluation is based on the tracking of a worker in an industrial setting.

3 Dataset description

The training datasets are provided as a `.csv` file, that can be loaded by various environments. Each line of the file contains a timestamp `rec_time ([float])` and a `json` string with one instance of measurement data. As mentioned above there will be two datasets for the different agents, where the significantly larger one is collected with a mobile robot as agent. The robot dataset (`training.csv`) does not contain the `rec_time ([float])` field, while both the experimental (`IPIN2022_T7_TestingTrial.csv`) and scoring datasets contain it. This means that `training.csv` only contains the `json` strings. The files contain the CI and reference positions. Each data instance (i.e. `json-string`) 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 (366 samples) corresponding to the imaginary and real parts of the CI in ns. (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. 2.

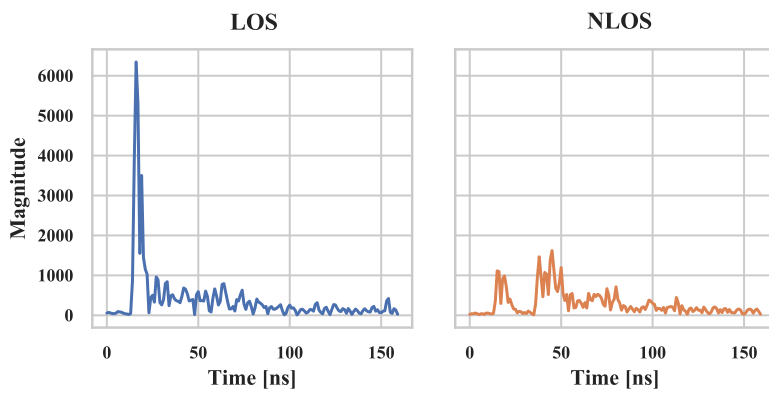

Figure 2: Visualization of two exemplary recordings in a LOS and NLOS case: the time labels of the x-axis are given in `ci_time`, the corresponding magnitudes on the y-axis are given by the complex numbered array defined by `ci_real` and `ci_imag`

.

- `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`.

Therefore the .json-string of each element has the form

```
{
    rec_time:  ...,
    ci_time:   [...],
    burst_id:  ...,
    ci_real:   [...],
    ci_imag:   [...],
    anch_id:   "..."
}
```

Fig. 2 shows two exemplary CI magnitudes, generated by combining the mentioned fields.

Again, for clarity, the data available at `https://owncloud.fraunhofer.de/index.php/s/YuUnhWiV9hvS7cw` are

- training.csv: The training data, obtained with the mobile robot, not containing `rec_time ([float])`
- IPIN2022_T7_TestingTrial.csv: A signficantly smaller amount of training data, obtained with a pedestrian/worker agent, containing `rec_time ([float])`. The same data is used for the experimental trial of the API (see later sections)

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.

3.1 Submission

The submission of results is done via the EvaalAPI (https://evaal.aaloa.org/evaalapi/), emulating a real-time localization setting. The general workflow for submission is as follows:

1. the user initially requests the latest data (i.e. the sensor readouts of the first 0.5 s) from the server, starting a new trial.
2. the user then estimates the position and sends it to the server. The server then advances the locally maintained time by 0.5 s and sends all sensor readouts that have occurred in this interval. This repeats until the trail ends or an error occurs.

You find details on the API and related communication in the online documentation on the website.

4 Challenge Objectives

For each setup (i.e. the trial) the initial position, perturbed by artificial additive zero-mean white Gaussian noise of standard deviation $1m$ in x and y-directions, is available: It is$[27.2, 9.8]\mathrm{m}$ for the experimental and $[26.0, 10.4]\mathrm{m}$ for the scoring trial.

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

For evaluation, the produced result trajectories we will resample to regular time intervals of 0.5 s using 1D-interpolation.

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. 3: 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.

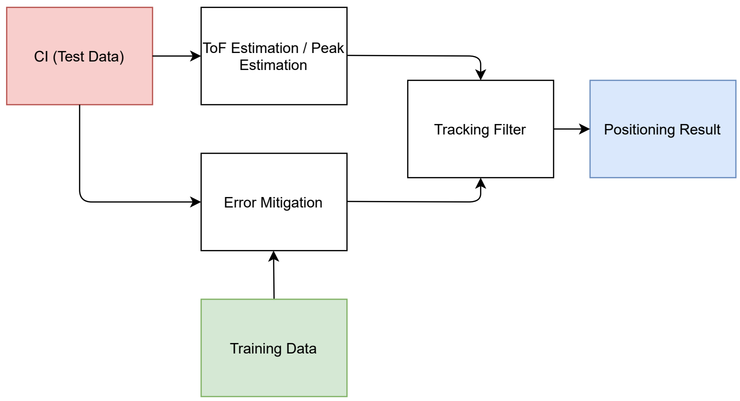

Figure 3: Exemplary Pipeline for a model error mitigation based system.

Fingerprinting: An exemplary positioning approach is sketched in Fig. 4: 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.

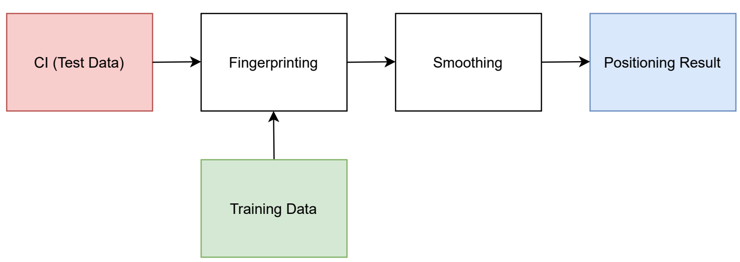

Figure 4: Exemplary Pipeline for a fingerprinting based system.

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. 5. 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.

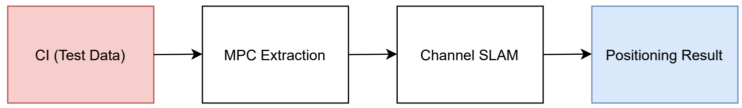

Figure 5: Exemplary Pipeline for a channel SLAM system.

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.

7 Download

You can download the training and validation datasets at `https://owncloud.fraunhofer.de/index.php/s/YuUnhWiV9hvS7cw`. Please don't hesitate contact us for any questions you might have.

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.