Multi-Sensor Integration for Sequential Monte Carlo Methods

Richard Hanten, Cornelia Schulz and Andreas Zell

There exist many Monte Carlo Localization approaches based on different sensor data, propagation or update models as well as map types. To speed up the development cycle of such methods, we propose a multi-sensor capable framework for Sequential Monte Carlo methods as well as a 2D Monte Carlo Localization instantiation. Thereby, we investigated abstraction, data synchronization and scheduling strategies which resulted in plugin-based and easily extensible implementations.

We evaluated the proposed concepts and our framework performance based on the Rawseeds dataset and were able to outperform state-of-the-art localization approaches. With our framework, new multi-sensor MCL approaches as well as other SMC methods could be developed easily.

I. Methods

All Sequential Monte Carlo methods, for example Monte Carlo Localization, share the same algorithmic core. The same three steps are applied to a SMC state, represented as a set of hypotheses, called particles: prediction $P$, update $U$ and resampling $R$.

To separate the SMC core from the applied functions, we use currying. This means that we bind propagation and update models to the necessary data and generate functions which only depend on the SMC sample set $X$.

To ensure correct propagation, we interpolate the control signals, e.g. wheel odometry data in case of MCL, at the desired update time steps.

To synchronize our input data streams, we wait for the most delayed input data and sort the updates by their correct time stamps. This is necessary, because e.g. stereo camera data is delayed by a stereo matching process. We call this strategy Lag Correction (LC).

To schedule the update functions, we use Completely Fair Scheduling (CFS), such that all update streams, e.g. sensor channels, get approximately the same overall execution time. Fig. 1 summarizes the structure of our framework for the proposed 2D MCL instantiation.

II. Results

For experiments and benchmarking, we used dataset 25a of Rawseeds [1]. The dataset contains data from a stereo camera and two LIDAR scanners.

We measured the impact of our scheduling and lag correction strategies in terms of drop rates for the individual sensor streams (‘FL’ for front laser, ‘RL’ for rear laser and ‘SC’ for stereo camera) as well as the localization accuracy (absolute trajectory error (ATE) and relative pose error (RPE)) we achieve with our framework, compared with AMCL [2].

Thereby, we used the AMCL default parameters, the likelihood field sensor model for the FL, the beam model for the RL and the ONDT model [3] for the SC to simulate different update runtimes.

III. Discussion

As can be seen in Tab. I, lag correction prohibits dropping all data from delayed input streams, in this case stereo camera data which is delayed by a stereo matching process. CFS scheduling ensures fair execution of different sensor models, such that less of the slow RL updates are executed and the overall localization accuracy improves. Finally, as illustrated in Tab. II, our framework outperforms AMCL, even with the same sensor models and parameters.

References