Bench-MR: Benchmarking Sampling-based Motion Planning Pipelines for Wheeled Mobile Robots

Eric Heiden*[1] Luigi Palmieri*[2] Leonard Bruns[3] Kai O. Arras[2] Gaurav S. Sukhatme[1] Sven Koenig[1]

[1] Dept. of Computer Science, University of Southern California, USA

[2] Robert Bosch GmbH, Corporate Research, Germany

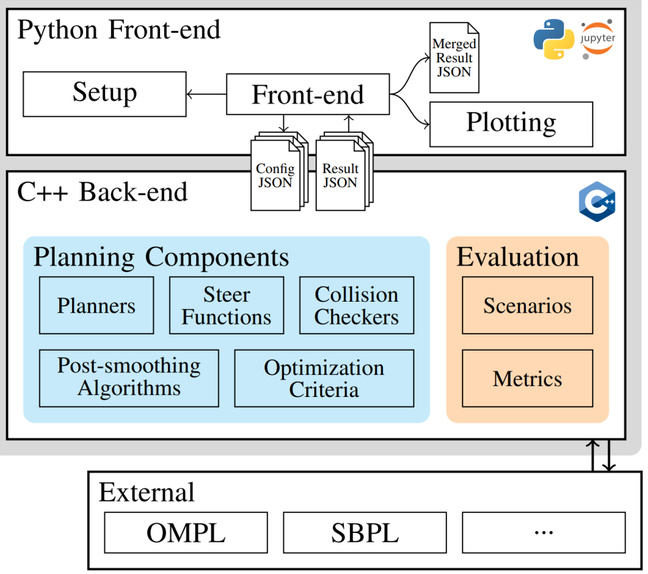
[3] Division of Robotics, Perception and Learning (RPL), KTH, Sweden

* denotes equal contribution

Features of Bench-MR

- Easy-to-use benchmarking software for motion planning components
- Applications in mobile robotics, intralogistics, autonomous driving
- Supports OMPL, SBPL as back-ends
- Various environment types, planners, extend functions, metrics

Architecture

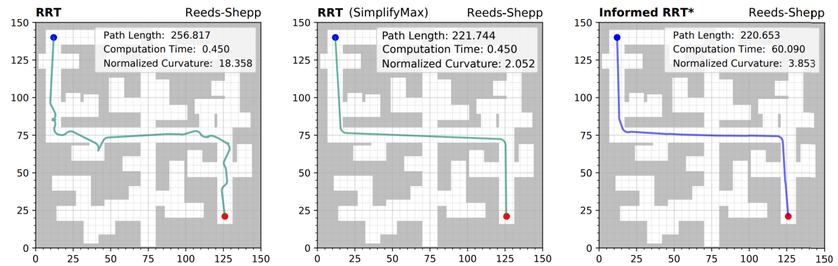


https://robot-motion.github.io/bench-mr

Exemplary Results

Combining feasible planners with post-smoothing techniques

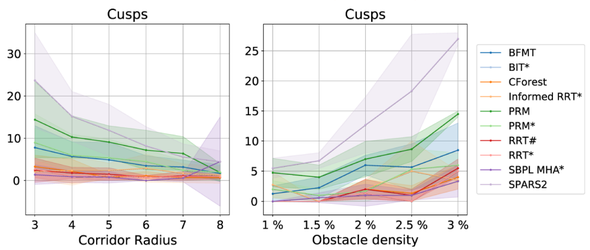
Achieve comparative performance to asymptotically optimal planners in significantly shorter time



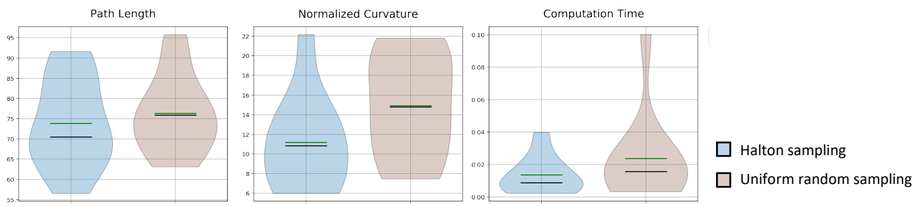
Feasible planner | Feasible planner + post-smoothing | Asymptotically optimal planner

Path quality in relation to environment complexity

Evaluate metrics over procedurally generated environments



Compare random and deterministic sampling



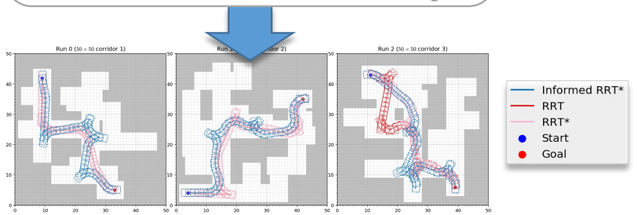
Front-end via Jupyter Notebooks

High-level user interface in Python to configure and run benchmarks, and analyze & plot the results

```
from mpb import MPB
mpb = MPB(config_file = 'benchmark_template.json')
mpb.set_corridor_grid_env(radius = 3)
mpb.set_planners(['rrt', 'rrt_star',
'informed_rrt_star'])
mpb.set_steer_functions(['reeds_shepp'])
mpb.run(runs=3)
mpb.visualize_trajectories()
```

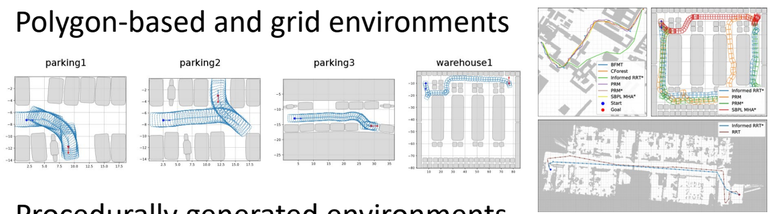
configuration of C++ back-end through JSON file

procedurally generated corridor-like environments



Environments

Polygon-based and grid environments



Procedurally generated environments