



# Motion Planning Test Suite

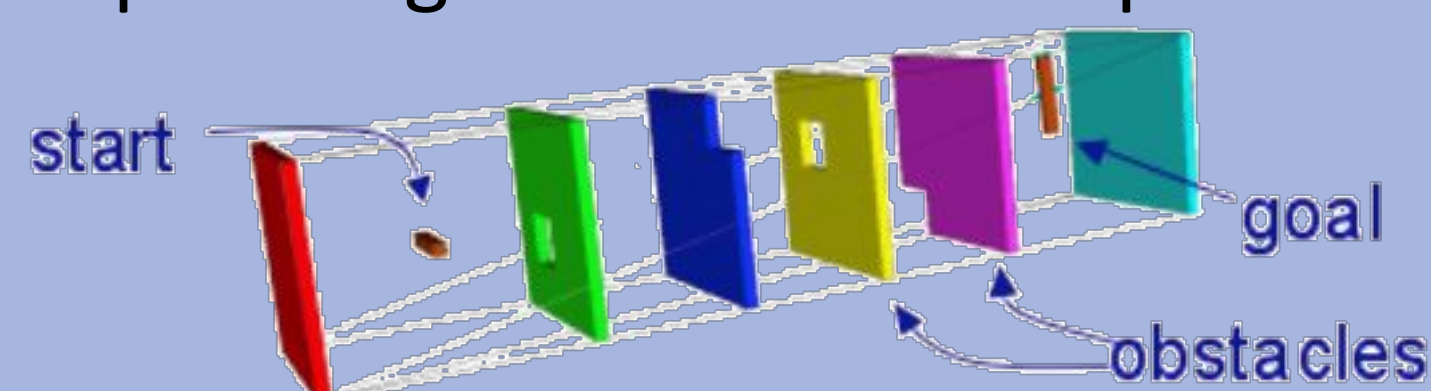
Jonathan Mulhern Art Martinez Marcos Pena Leonel Pena Brandon Martinez  
Irving Solis Shawna Thomas Nancy M. Amato

jmlhern@albany.edu art.martinez2329@gmail.com mpena16123@gmail.com leopena117@gmail.com b.m\_755@hotmail.com g02005760@cse.tamu.edu sthomas@cse.tamu.edu amato@cse.tamu.edu



## What is Motion Planning?

- The problem of finding a valid path for a movable, non-static object from a start to a goal configuration.
- Motion planning is an intractable problem.



## Motivation

- Suite designed to evaluate motion planning algorithms.
- Helps Parasol Lab exercise their algorithms in a range of scenarios.
- The goal is to provide a set of comprehensive problems.



## Test Suite Objective and Requirements

The test suite is a set of scenarios that serve as benchmarks for future work. Each benchmark was designed to analyze different motion planning strategies with a set of environments, and therefore have a test tool. Representative set of problems that covers Parasol's interests.

### PMPL

- C++ motion planning library developed by the Parasol Lab.
- Capable of solving different kinds of motion planning problems by using sampling based techniques.
- Sampling methods include: MAPRM, OBPRM, RRT, Bridge, etc.

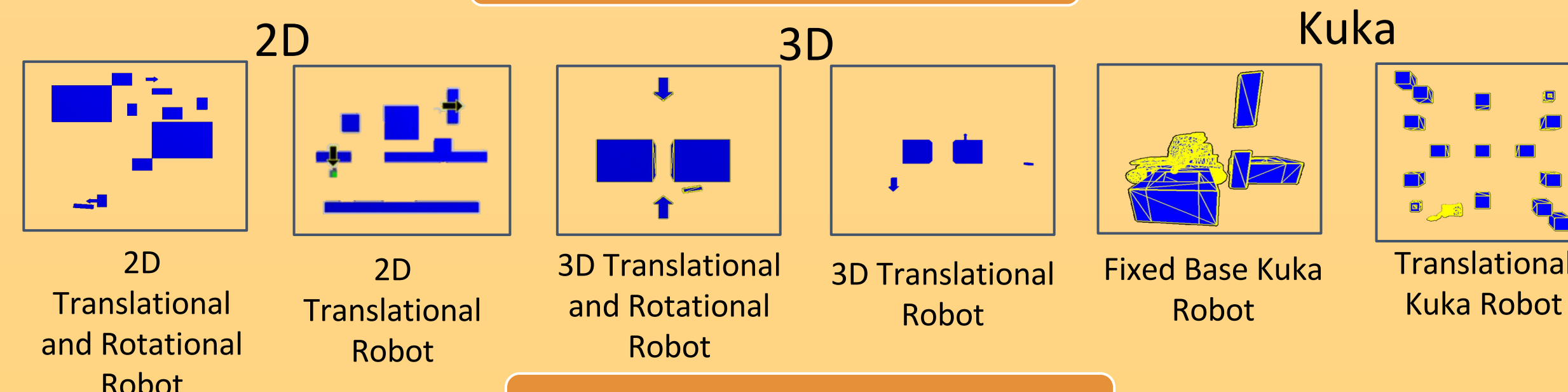
### Vizmo++

- Visualization tool for reading and showing a PMPL solution.
- Main outputs are the sampling process, a roadmap, and a query.
- Can't process physical variables as gravity, friction, and inertia forces.

### Simulator

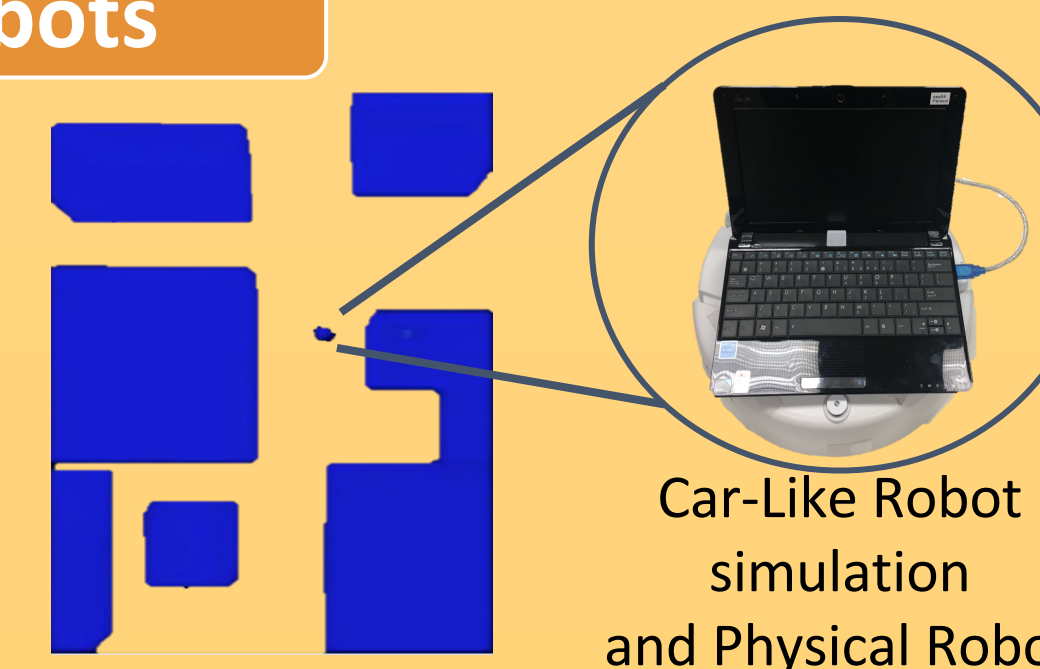
- Shows the result when a PMPL plan is applied to a robot under real world characteristics.
- Able to connect to a physical robot for implementing a PMPL plan in the real world.

### Simulated Robots



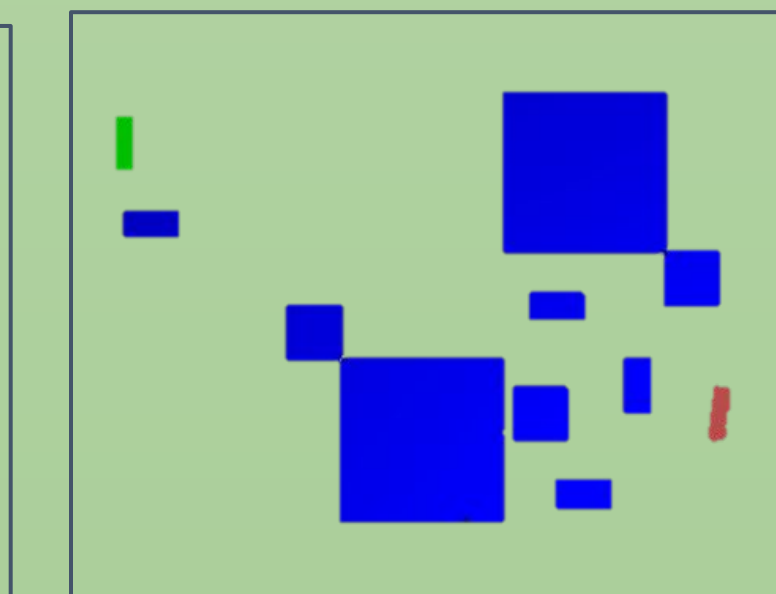
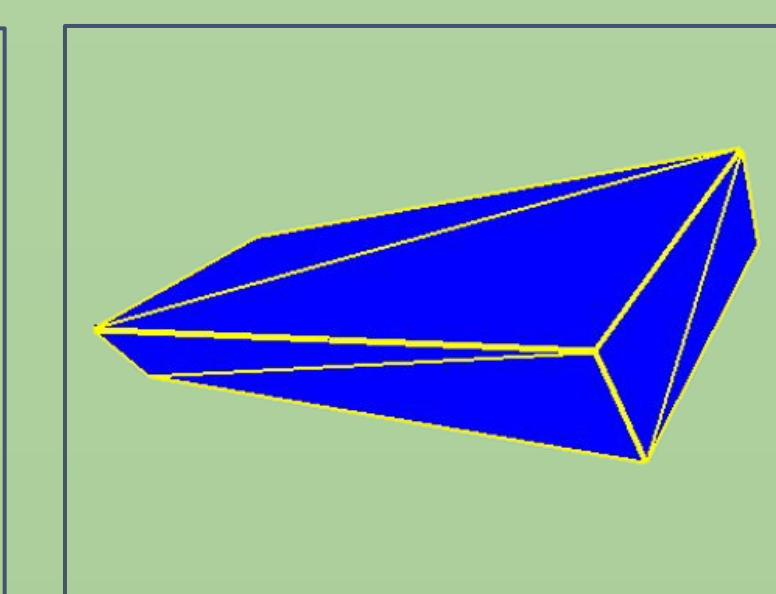
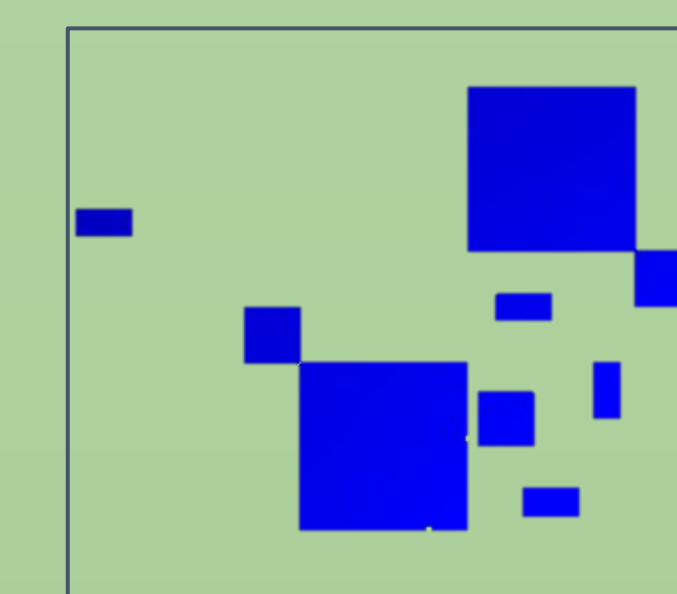
### Physical Robots

- Robot with a netbook mounted atop.
- Receives a path file converted into translation, rotation, and timing commands.
- Car-like movement with specific set of possible movements.



## Benchmark Development Process

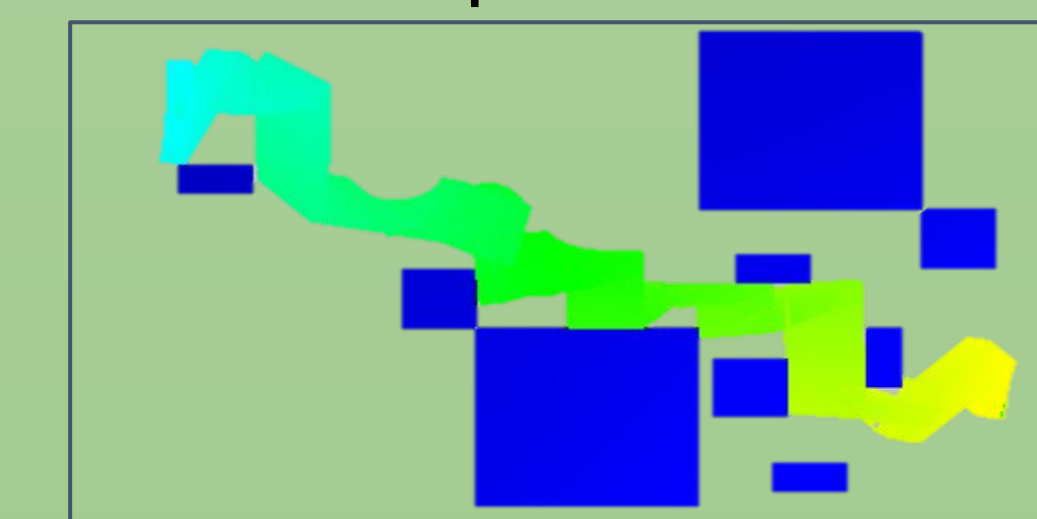
- An Environment is created
- A Robot is created
- A Query is created



- Motion Planning Strategy is applied

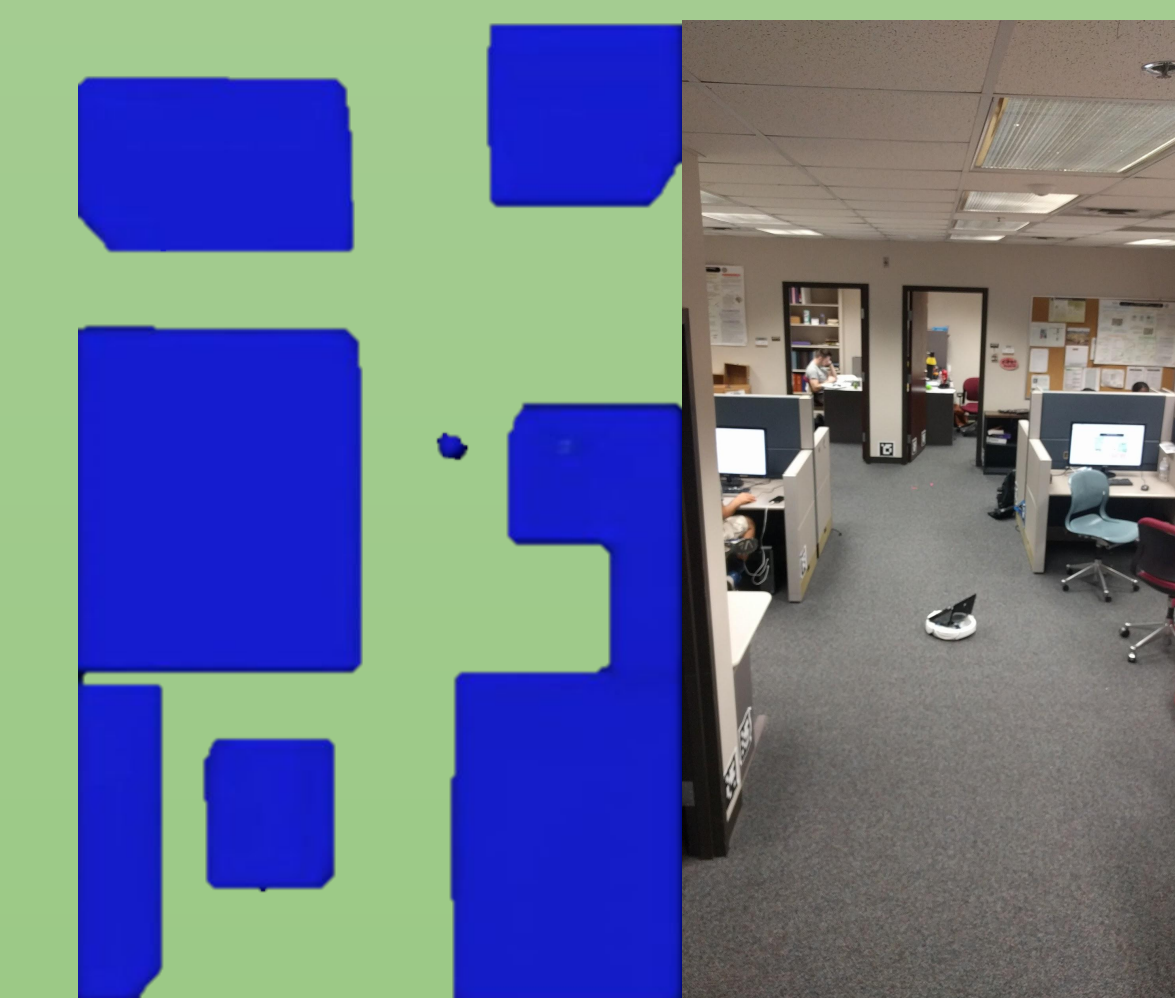


- Robot follows final path



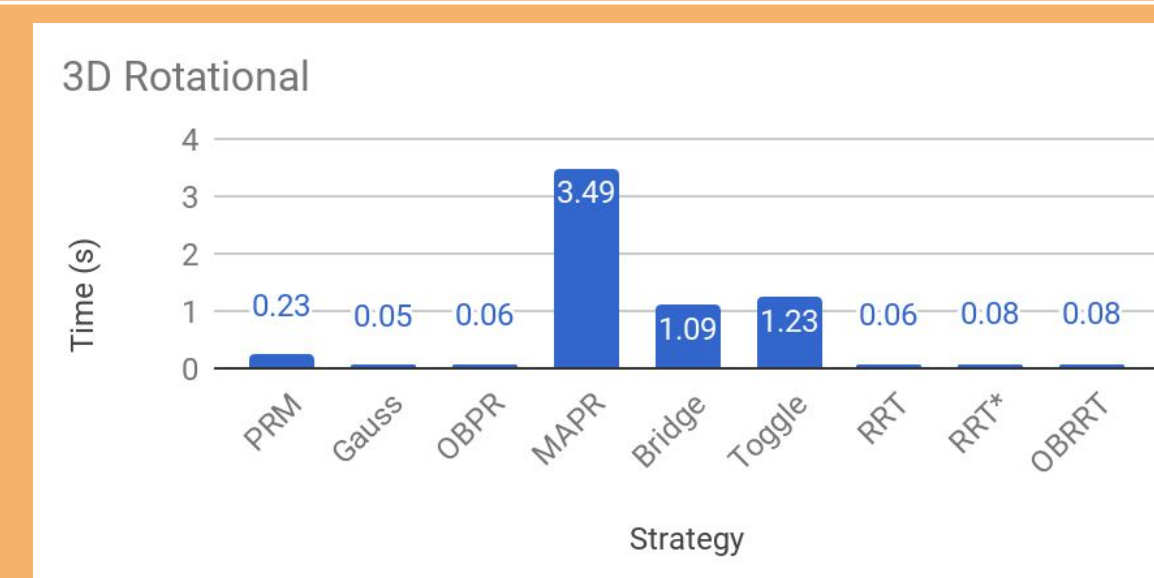
For Physical Robots:

- Define controls for the robot
- Connection is established between Robot and Host
- Robot follows simulation



## Benchmark Statistics

Here are our statistics for the test suite. Note that these aren't results, but rather statistics portraying that the benchmarks work and that they are operating correctly with the mentioned strategies. With the exception of MAPRM on the translational Kuka benchmark, all strategies performed as expected. We believe the reason MAPRM did not succeed in the translational Kuka benchmark was due to the high complexity in finding the medial axis for high degrees of freedom robots, which the translational Kuka had 8 DoF's. This in turn took so long that we never received results.



## Conclusion

The simulator was great, however, it may need some improvements in terms of its similarity to Vizmo++. While there was less than a 1% difference between Vizmo and the suite, it still needs some work, as bugs do exist, such as the reset not working, needing more options in the UI, or providing an interface for environment modification. Being able to handle dynamic constraints and changing environments is an ideal future goal. Overall, this is a great step forward into providing more tools for the Parasol motion planning group.

## Acknowledgements

We want to give thanks to the DREU, USRG, Texas A&M, Parasol Labs, and the NSF. We also want to thank our mentors, Dr. Nancy Amato and Irving Solis, and all employers within Parasol Labs. This research was supported by the Department of Computer Science and Engineering at Texas A&M University [and supported in part by the CRA-W Distributed REU (DREU) project].