An Analysis of Motions in Microseconds

Pravi Samaratunga

Boston University ECE

September 5, 2024

Presentation Overview

- 1. Motivation
- 2. Sampling-Based Motion Planning
- 3. Motion Primitives
- 4. *Motions in Microseconds*
- 5. Strengths & Weaknesses
- 6. Future Work

Motivation

(source: https://spectrum.ieee.org/falling-robots)

Sampling-Based Motion Planning

Sampling-Based Motion Planning

What is Sampling-Based Motion Planning (SBMP)?

Sampling-based motion planning SBMP is a method of determining how a robot moves through configuration space by sampling values from that space.

Sampling-Based Motion Planning

What is Sampling-Based Motion Planning (SBMP)?

Sampling-based motion planning SBMP is a method of determining how a robot moves through configuration space by sampling values from that space.

Why is it relevant?

SBMP algorithms are widely applicable and used in many fields. However, it can take several seconds for SBMP methods to execute, so there is ample space for innovation.

Sampling-Based Motion Planning

What is Sampling-Based Motion Planning (SBMP)?

Sampling-based motion planning SBMP is a method of determining how a robot moves through configuration space by sampling values from that space.

Why is it relevant?

SBMP algorithms are widely applicable and used in many fields. However, it can take several seconds for SBMP methods to execute, so there is ample space for innovation.

want		have
	$\leftrightarrow$	
< 1ms		many seconds

Sampling-Based Motion Planning

What is Sampling-Based Motion Planning (SBMP)?

Sampling-based motion planning SBMP is a method of determining how a robot moves through configuration space by sampling values from that space.

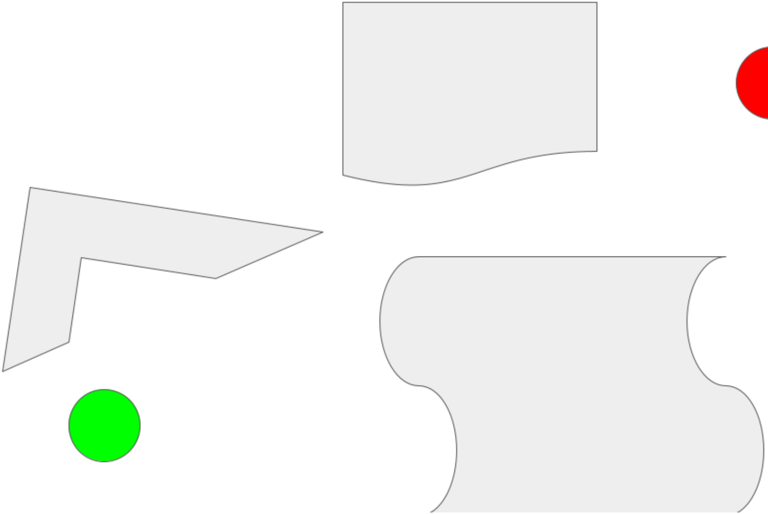
Why is it relevant?

SBMP algorithms are widely applicable and used in many fields. However, it can take several seconds for SBMP methods to execute, so there is ample space for innovation.

How do we optimize SBMP?

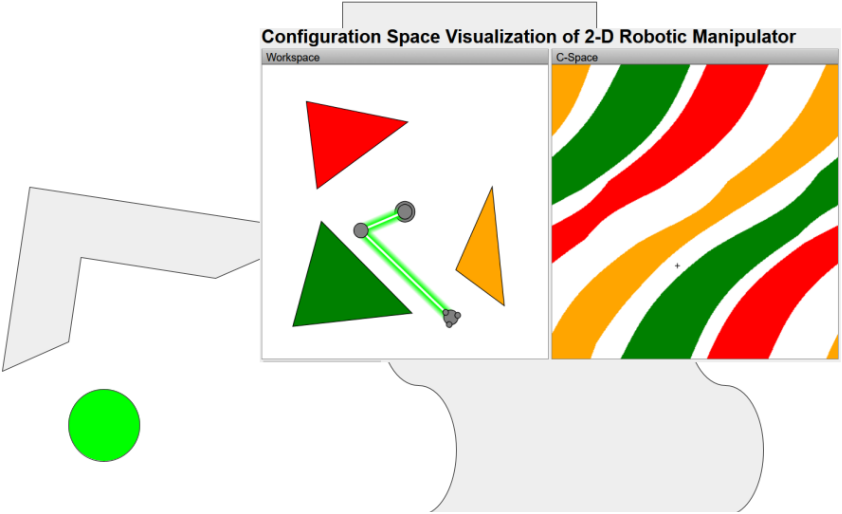
Over the years, many optimizations have been introduced to SBMP. We will discuss VAMP, the Vector-Accelerated Motion Planning system as introduced by Thomason, Kingston, and Kavraki 2023.

Example SBMP Execution

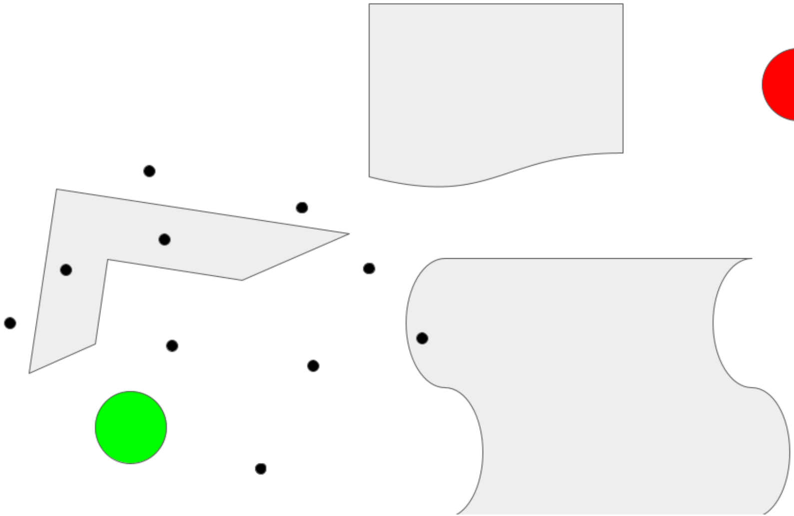


Problem Definition

Example SBMP Execution

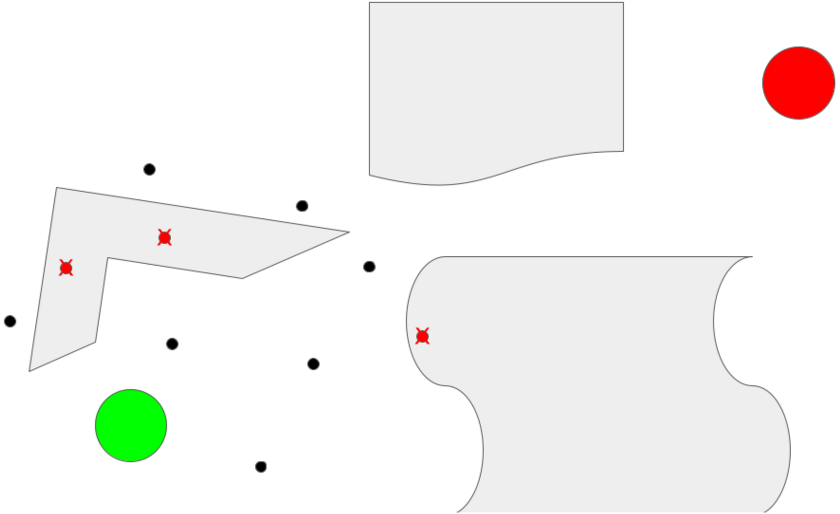


Example SBMP Execution



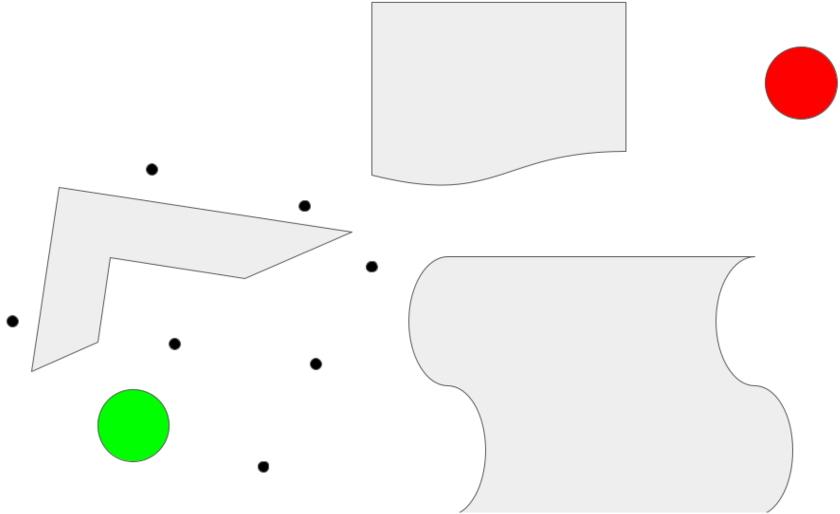
Sampling

Example SBMP Execution



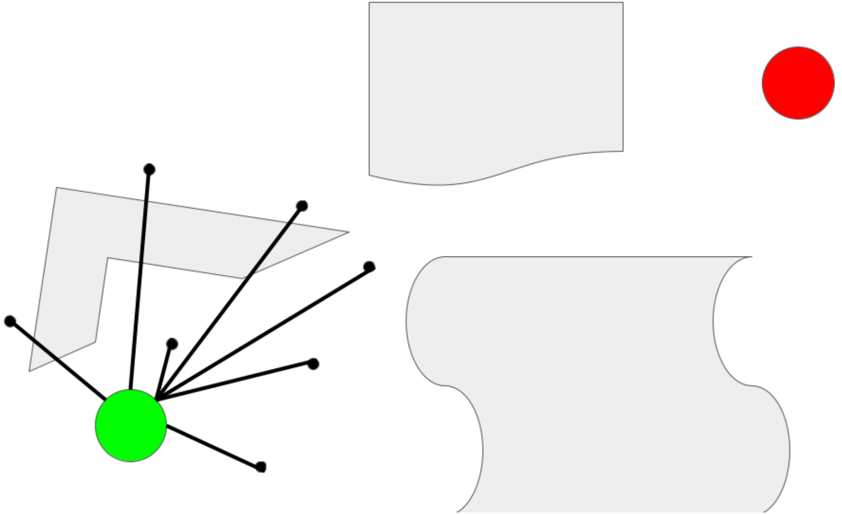
State Collision Check

Example SBMP Execution



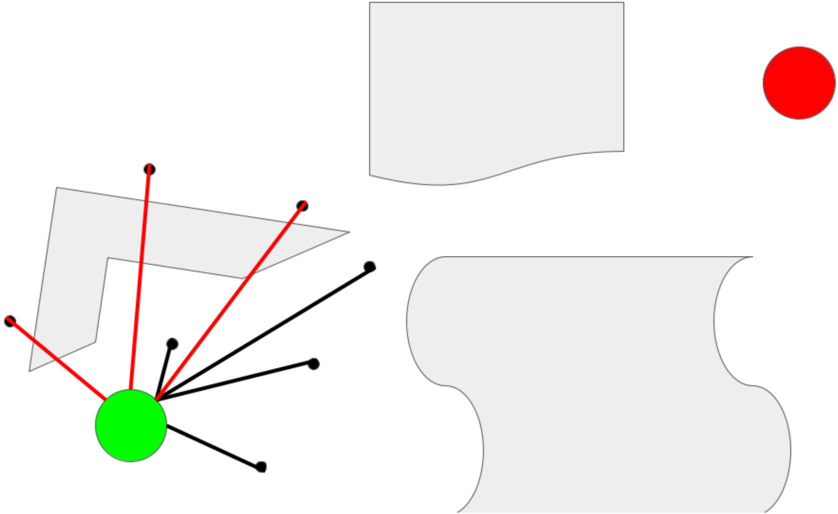
State Validation

Example SBMP Execution



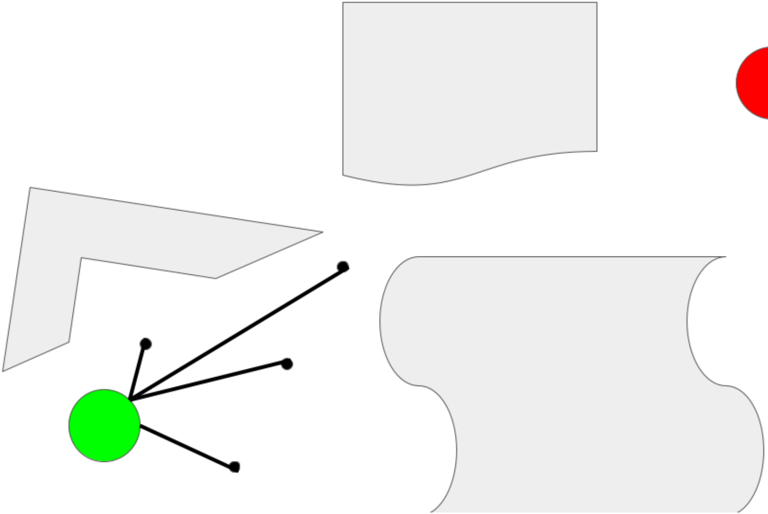
Local Planning

Example SBMP Execution



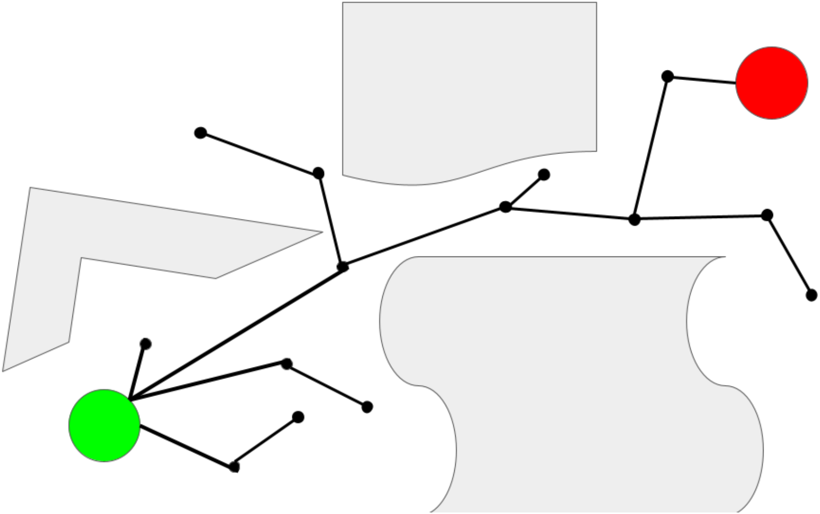
Edge Collision Check

Example SBMP Execution



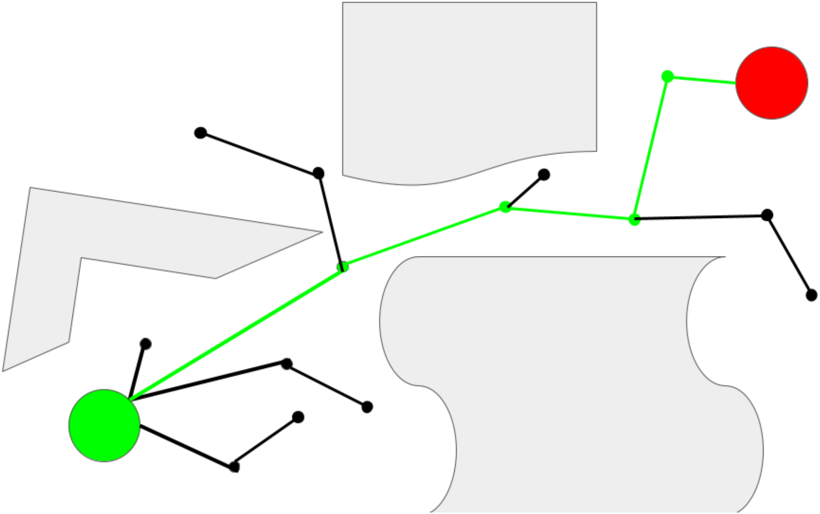
Edge Validation

Example SBMP Execution



Generate Graph

Example SBMP Execution



Find Path

Planning Primitives

Planning Primitives

Nearest Neighbors Search (NN)

Part of the sampling function for the configuration space.

Forward Kinematics (FK)

Maps configuration-space elements to physical space.

Collision Checking (CC)

Ensures that obstacles do not collide with the robot. Uses many passes of FK in computation.

CC is conventionally thought of as the most computationally intensive step of the motion planning process, taking up to 90% of planning time. [Bialkowski, Karaman, and Frazzoli 2011]

How do we optimize these computationally intensive tasks?

State of the Art

State of the Art

- Motion planning parallelism is typically acheived by running several independent planners in parallel.

State of the Art

- Motion planning parallelism is typically acheived by running several independent planners in parallel.
- Hardware acceleration of motion planning has been explored, but necessarily introduces latency when communicating with the CPU.

State of the Art

- Motion planning parallelism is typically acheived by running several independent planners in parallel.
- Hardware acceleration of motion planning has been explored, but necessarily introduces latency when communicating with the CPU.
- FK computations are difficult to parallelize due to the robot geometry introducing data dependencies.

State of the Art

- Motion planning parallelism is typically acheived by running several independent planners in parallel.
- Hardware acceleration of motion planning has been explored, but necessarily introduces latency when communicating with the CPU.
- FK computations are difficult to parallelize due to the robot geometry introducing data dependencies.
- In order to minimize the total number of CC calls, it is often performed in two steps, *broadphase* and *narrowphase*. Shah, Yang, and Aamodt 2023 developed a method for using local sparsity patterns in a parallel manner.

Vector Accelerated Motion Planning (VAMP)

VAMP presents a novel planner that uses *fine-grained* instruction-level parallelism to achieve substantial speedups.

Vector Accelerated Motion Planning (VAMP)

VAMP presents a novel planner that uses *fine-grained* instruction-level parallelism to achieve substantial speedups.

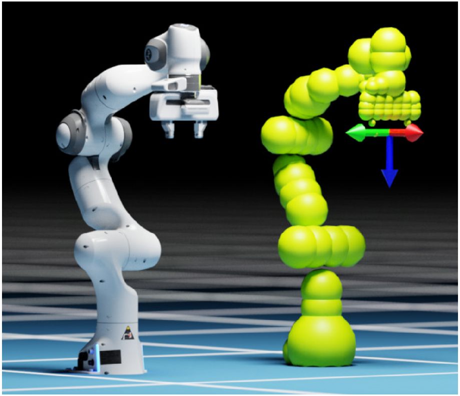
VAMP's innovations over State of the Art:

- 1. Geometric Intersection Tests
- 2. Tracing Compiler
- 3. Struct-of-Arrays data representation
- 4. Raked Motion Validator implementation

Geometric Intersection Tests

By representing the robot as a system of geometric objects, intersection tests between pairs of objects can be vectorized.

This approach does away with the staged broadphase and narrowphase approach to collision checking, but focuses soley on narrowphase.



[Sundaralingam et al. 2023]

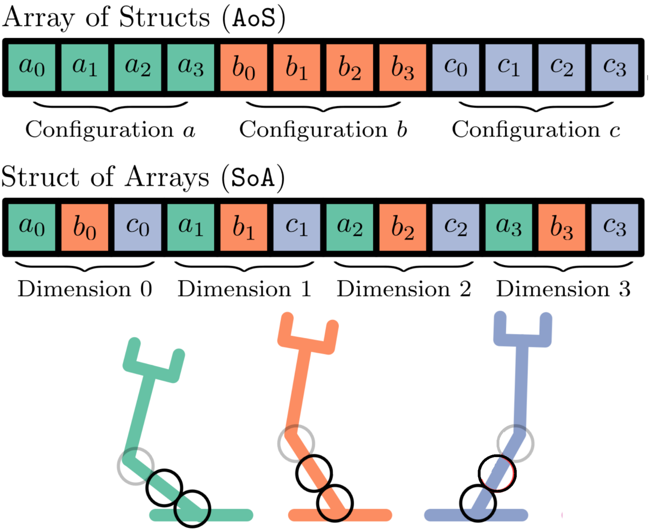
Tracing Compiler

To vectorize FK:

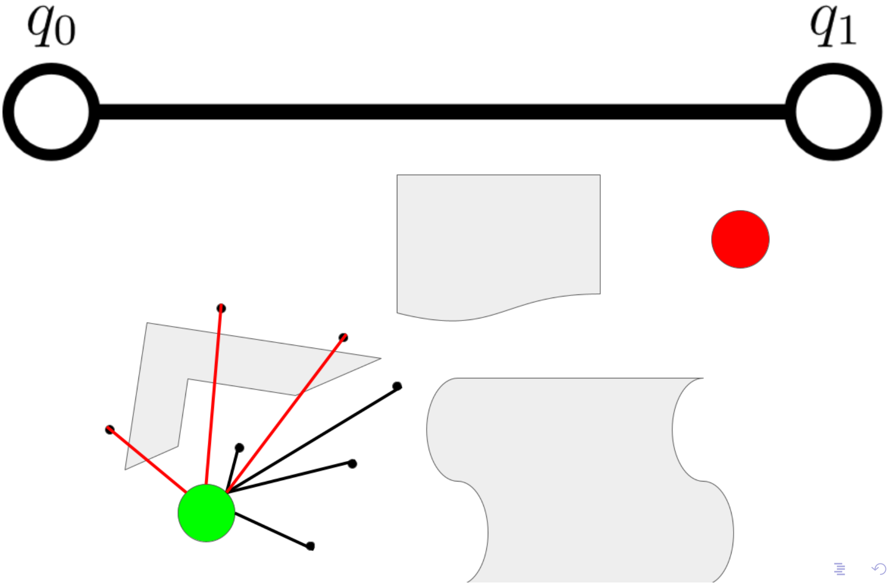
- Trace the operations of the robot's kinematics from a URDF file.
- Generate a vector configuration structure for a batch of configurations.
- Output a minimal set of operations to compute the traced function as an unrolled loop.

Notably, this tracing technique generates optimized FK implementations at compile time through automatic code generation.

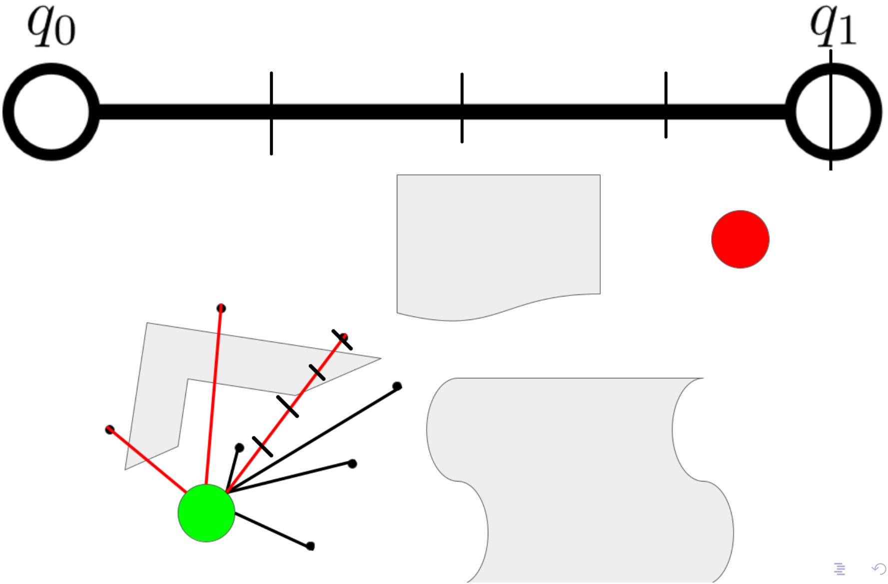
Struct-of-Arrays



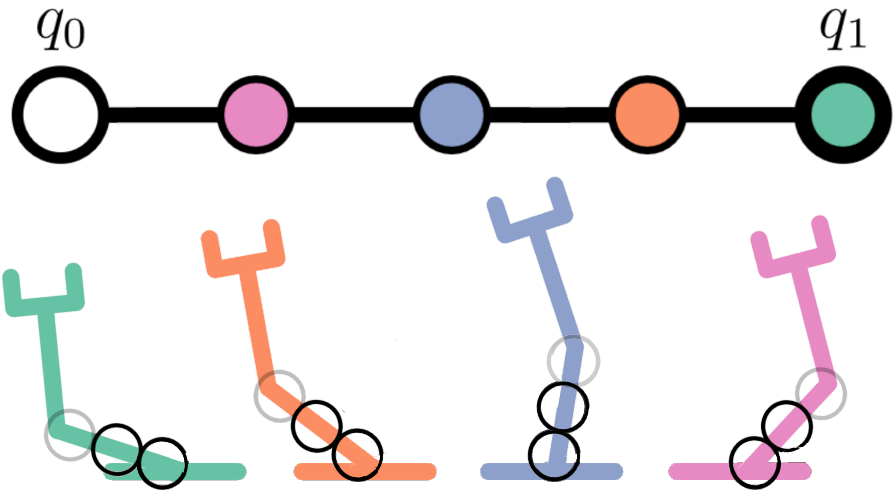
Raked Motion Validator



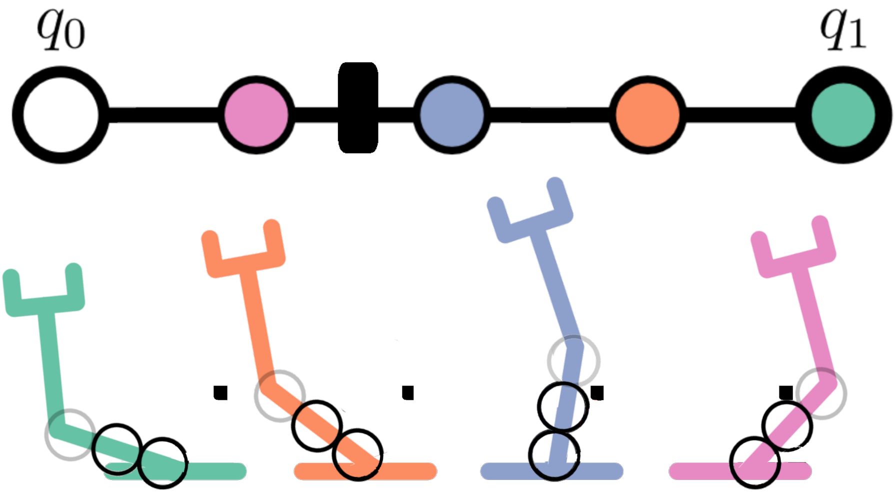
Raked Motion Validator



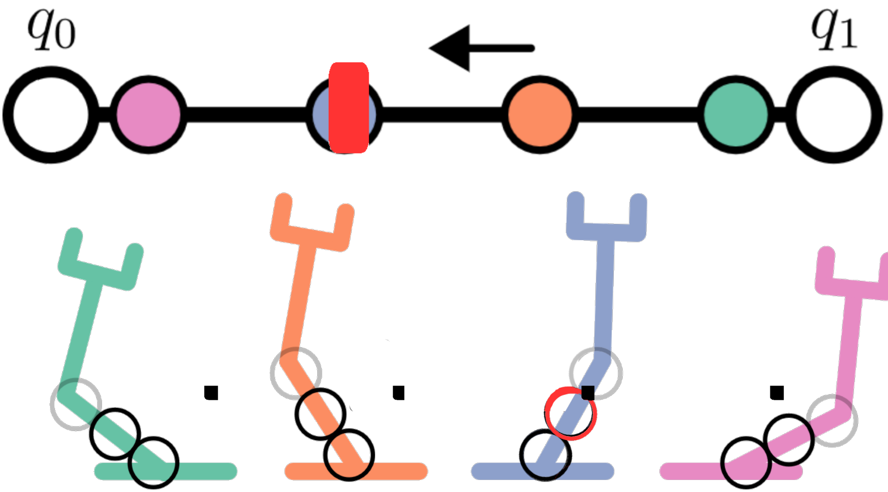
Raked Motion Validator



Raked Motion Validator



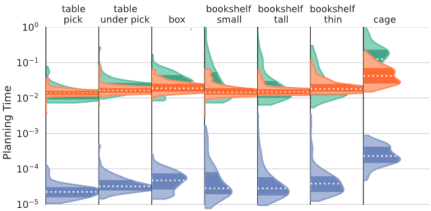
Raked Motion Validator



Strengths

Motions in Microseconds represents serious improvements over State of the Art.

With straightforward architechtural optimizations, the authors get upwards of 500× speedups over PyBullet, and 100-200× over MoveIt, both using OMPL for the motion planning implementation.



Weaknesses

The authors don't show an ablation study of which optimizations make which end-to-end performance impacts.

Weaknesses

The authors don't show an ablation study of which optimizations make which end-to-end performance impacts.

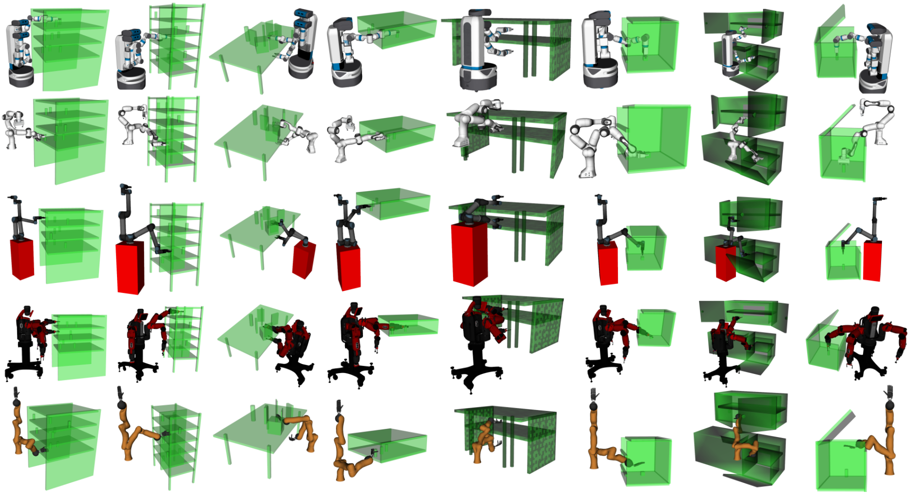
The paper does not describe what impacts the optimizations have as the problem size scales.

	vs VAMP	Mean	Q1	Median	Q3	95%
Panda (7 DOF)	PyBullet	$\times 403.03$	$\times 797.69$	$\times 888.46$	$\times 786.02$	$\times 360.09$
	MoveIt	$\times 55.36$	$\times 2594.63$	$\times 1126.30$	$\times 247.47$	$\times 28.35$
Fetch (8 DOF)	PyBullet	$\times 108.01$	$\times 314.31$	$\times 434.04$	$\times 212.13$	$\times 86.64$
	MoveIt	$\times 13.39$	$\times 60.95$	$\times 34.47$	$\times 16.51$	$\times 11.86$

Avenues for Future Work

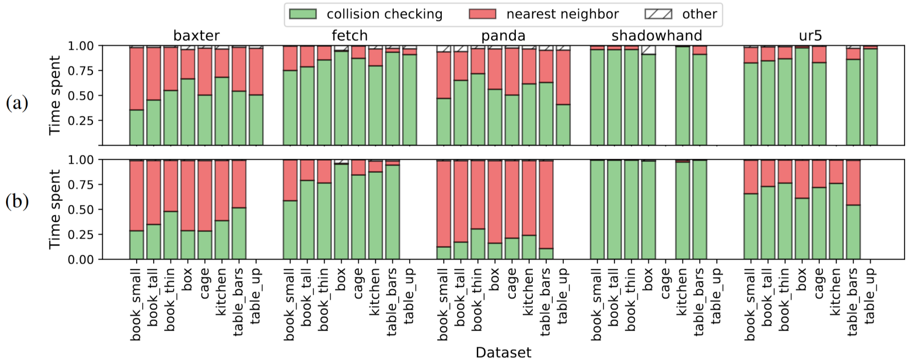
- Measure and identify precise bottlenecks within SBMP
- Optimize geometric intersection tests further

Measuring Bottlenecks



[Chamzas et al. 2022]

Measuring Bottlenecks



[Ghosal, Sacher, **Samaratunga**, Neuman, Plancher, Reddi 2024 (in preparation)]

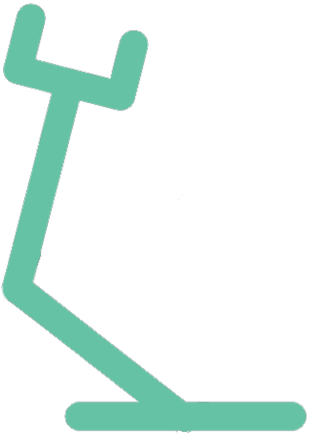
Bounding Volume Fidelity Research

- Exploit structured information about manipulator & environment
- Explore non-sphere geometric representations of colliders (eg trimeshes, capsules)
- Explore bounding volume hierarchy fidelity

[Sundaralingam et al. 2023]

End of Presentation

Motions in Microseconds presents novel optimizations for SBMP.
The main contribution, VAMP is substantially faster than the state of the art.
While the results are extremely promising, there is still further space for optimization.



References I

- Bialkowski, Joshua, Sertac Karaman, and Emilio Frazzoli (2011). "Massively parallelizing the RRT and the RRT". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3513–3518. DOI: 10.1109/IROS.2011.6095053.
- Chamzas, Constantinos et al. (Apr. 2022). "MotionBenchMaker: A Tool to Generate and Benchmark Motion Planning Datasets". In: *IEEE Robotics and Automation Letters* 7.2, pp. 882–889. ISSN: 2377-3774. DOI: 10.1109/lra.2021.3133603. URL: http://dx.doi.org/10.1109/LRA.2021.3133603.

References II

Orthey, Andreas, Constantinos Chamzas, and Lydia E. Kavraki (2024). "Sampling-Based Motion Planning: A Comparative Review". In: *Annual Review of Control, Robotics, and Autonomous Systems* 7.Volume 7, 2024, pp. 285–310. ISSN: 2573-5144. DOI: https://doi.org/10.1146/annurev-control-061623-094742. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-061623-094742.

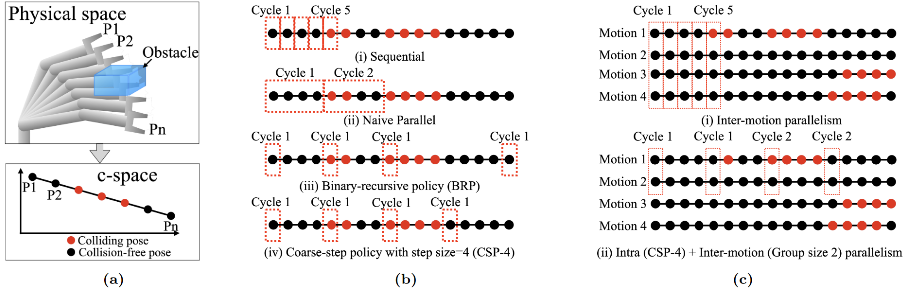
Shah, Deval, Ningfeng Yang, and Tor M. Aamodt (2023). "Energy-Efficient Realtime Motion Planning". In: *Proceedings of the 50th Annual International Symposium on Computer Architecture*. ISCA '23. Orlando, FL, USA: Association for Computing Machinery. ISBN: 9798400700958. DOI: 10.1145/3579371.3589092. URL: https://doi.org/10.1145/3579371.3589092.

References III

Sundaralingam, Balakumar et al. (2023). *cuRobo: Parallelized Collision-Free Minimum-Jerk Robot Motion Generation*. arXiv: 2310.17274 [cs.RO]. URL: https://arxiv.org/abs/2310.17274.

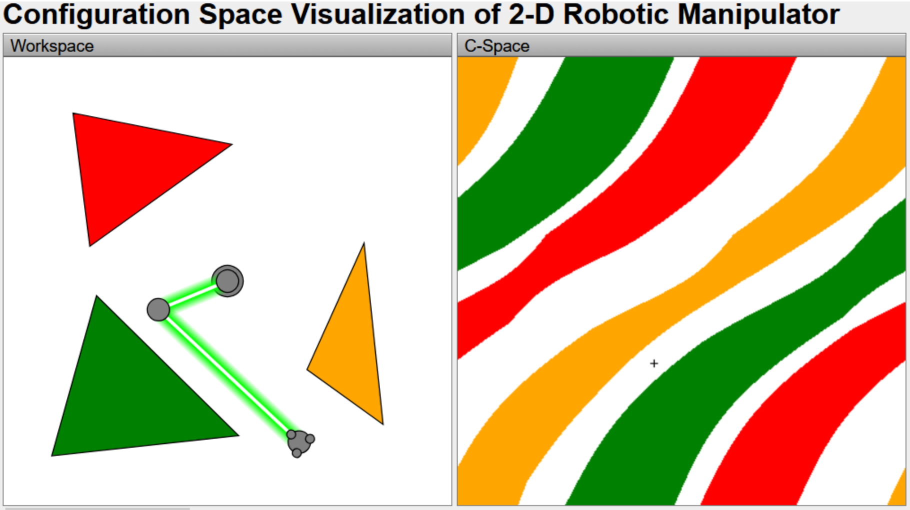
Thomason, Wil, Zachary Kingston, and Lydia E. Kavraki (2023). *Motions in Microseconds via Vectorized Sampling-Based Planning*. arXiv: 2309.14545 [cs.RO]. URL: https://arxiv.org/abs/2309.14545.

Raked Motion Validator Implementation



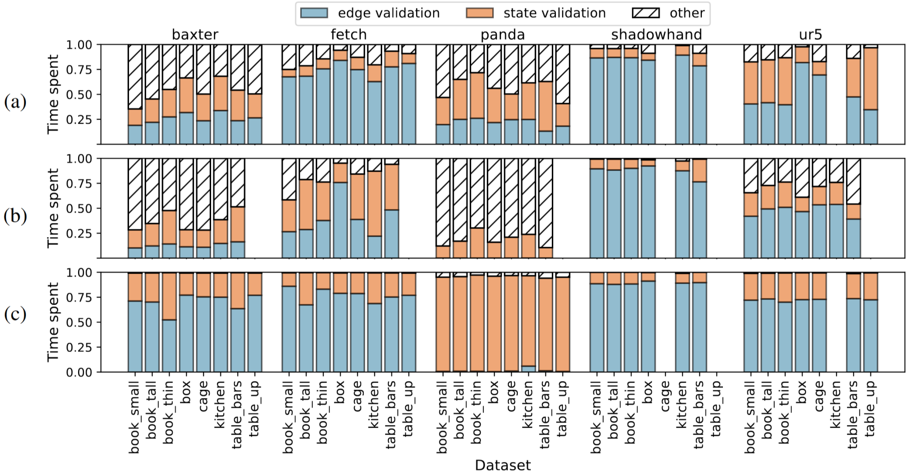
[Shah, Yang, and Aamodt 2023]

2D Configuration Space Visualization

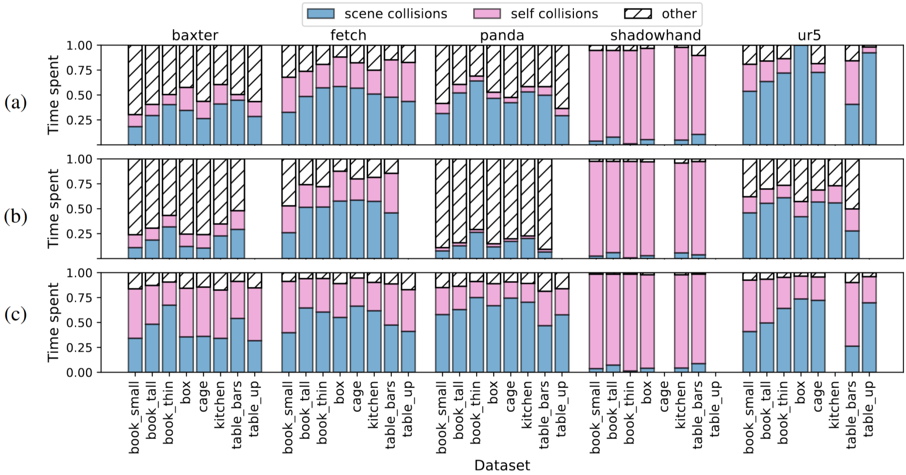


[https://www.cs.unc.edu/ jeffi/c-space/robot.xhtml]

Edge Validation vs State Validation



Scene Collisions vs Self Collisions



Further Credits

1. Orthey, Chamzas, and Kavraki 2024 for concise motivation description.
2. Sundaralingam et al. 2023 for spherification figures