Introduction to State-of-the-art Motion Planning Algorithms

Presented by
Konstantinos Tsianos
Robots need to move!
Motion

- Robot motion must be continuous
- Geometric constraints
- Dynamic constraints
- Safety constraints
- Execution constraints
Path planning – A to B paradigm
A real robot has a shape
Given its geometry, a robot is fully specified by a minimal set of parameters $p_1, p_2, \ldots, p_n$
- Generalized coordinates
- Degrees of freedom
- Configurations space parameters

If $p_i \in X_i$, the configuration space (C-Space) is a manifold

$$Q = X_1 \times X_2 \times \cdots \times X_n$$
Configuration space of a 2-link arm
Some more examples

2-link planar arm among 5 obstacles

3-link planar arm

workspace

C-space
Path Planning

Given:
- World geometry
- Robot’s geometry
- Start and goal configuration

Compute:
A collision-free, feasible path to the goal
Complete planners - Examples

Visibility roadmap

Voronoi diagram

Trapezoidal decomposition
Path planning is hard.

Understanding the structure of C-spaces is a very hard mathematical challenge.

C-spaces for most interesting problems are high dimensional.

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>COMPLEXITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sofa Mover (3DOF)</td>
<td>O(n^{2+\varepsilon}) - not implemented [HS96]</td>
</tr>
<tr>
<td>Piano Mover (6DOF)</td>
<td>Polynomial – no practical algorithm [SS83]</td>
</tr>
<tr>
<td>n Disks in the Plane</td>
<td>NP-Hard [SS83]</td>
</tr>
<tr>
<td>n Link Chain in 3D</td>
<td>PSPACE-Complete [HSS87]</td>
</tr>
<tr>
<td>Generalized Mover</td>
<td>PSPACE-Complete [Canny88]</td>
</tr>
</tbody>
</table>
Potential field planners

\[ U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q) \]

\[ F(q) = -\nabla U(q) \]
Local minima problem
• Exact navigation functions are mathematically challenging in general (RK91)

• Approximate navigation functions are intractable in many dimensions
Planning does not require understanding the structure of the configuration space!
• Try to capture the connectivity of the C-Space without explicitly constructing it
• Use sampling – possible due to development of fast collision checking algorithms
• Result in very efficient algorithms that showed solutions to many previously intractable geometric planning problems
Two difficult examples

Alpha Puzzle 1.0 Solution
James Kuffner, Feb. 2001

model by DSMFT group, Texas A&M Univ.
original model by Boris Yamrom, GE
Sampling-based planning

Sampling Function

Motion Planner

Collision Checker

Workspace Model

Robot Model

Collision-free Path
Probabilistic Roadmap (PRM)
Probabilistic Roadmap (PRM)
Probabilistic Roadmap (PRM)
Probabilistic Roadmap (PRM)
Probabilistic Roadmap (PRM)
Probabilistic Roadmap (PRM)
Key components of the planner

- Nearest neighbor datastructures
- Sampling strategy [AB98, HR03, OS99, BL01]
- Connection strategy
- Collision checking [LM97, LM03]
- Graph search

![Diagram of key components]

**Sampling Function**

**Motion Planner**

**Collision Checker**

- Workspace Model
- Robot Model

Collision-free Path
Tree-based planner (RRT)
Tree-based planner (RRT)
Tree-based planner (RRT)
Tree-based planner (RRT)
Tree-based planner (RRT)
• Rapidly-exploring Random Tree (RRT) [KL99, LK01]
• Expansive Space Tree (EST) [H97, H00]
• Single Query Bidirectional Lazy Tree (SBL) [SL01]
• Guided Expansive Space Tree [PK03]
• Adaptive Dynamic Domain RRT [Y05]
• Utility-guided RRT [BB07]
• Particle RRT [NR07]
• …
• Real robots have dynamic and actuation constraints
• Tree-based planners can easily accommodate a forward dynamics simulator to plan under realistic motion constraints
• Trees are directed data structures that nicely encode the notion of time
• Propagation is control based
Path planning for a car (PRM)
Planning with velocity control (IST)
Planning with acceleration control (IST)
Building a tree with IST
More examples from the latest planners

- DSLX
- KPIECE
- Real-time replanning
- Real-time multirobot exploration
Combining discrete search with continuous exploration
Planning in high dimensional state and control spaces
Real-time replanning

Planning in partially known environments within moving obstacles
Multirobot coordinated exploration
Some facts about sampling-based planners

- General and applicable to many different systems
- Very efficient on average, yielding solutions to many previously intractable problems
- Relatively simple to describe and implement
- Subject to problem specific optimizations
  - At best only probabilistically complete
  - Face difficulties in dealing with narrow passages
• Sampling-based planners contain many standardized building blocks
• As planners get more complicated it is useful to not have to code the basic components again and again
• There is not common benchmarking platform and planning problems
• Performance comparisons are not always fair
• Speedups may be misleading due to non-uniform implementation details
• Sampling-based planners contain many standardized building blocks
• As planners get more complicated it is useful to not have to code the basic components again and again
• **There is not common benchmarking platform and planning problems**
• Performance comparisons are not always fair
• Speedups may be misleading due to non-uniform implementation details
Where OOPSMP comes in

- Different planners can be tested on the exact same planning scenario
- All planners use common data structures and utilities
- “Quick and dirty” testing is possible due to many already existing planners and modules
- The focus can be on algorithmic aspects rather than implementation details
- Can be integrated with different visualization software and physics simulators for nicer looking and more realistic results
- OOPSMP is currently unique at providing a large variety of sampling-based motion planners, data structures and building blocks that are tuned for motion planning applications
Thank you!