RMP-Sample Based Motion Planning

Key words	connectivity, expansiveness, mulitple query, probabilistic roadmap, single query, smoothing
Status	note complete

Dis

Combinatorial approaches:

- Advantages: **complete algorithm**
- Disadvantages:
  - **computational expensive**, DOF ↑, runtime increases exponentially ↑
  - construction of **whole configuration space**(collision tests) —> for larger DOF very expensive
  - uses heuristics, not reliable. (potential field)

—> Solution: **Sample based methods!!!**

Sample based Methods

- **probabilistic completeness:** a solution path can be **found with high probability**.

▼ Distribution of sample points: **uniform random distribution**

- on which parameters?

- **Parametrization** of environment to get **a connected roadmap**

Probabilistic Roadmap (PRM) — Multiple-query sampling for higher dimensions solving multiple-query problems

- **multiple query sampling**
- assumptions:
  - **static** environment, arbitrary obstacles
  - **many queries** processed in same environment
  - **unknown start & goal**, can be anywhere in free space
  - repeated tasks

  —> **sample over the whole free space** necessary!!
- applications:
  - navigation in static environment
  - robot manipulator arm in a workcell (parts from different start, need to be moved to different goals)
  - playing chess
- Process:
  - **Offline: Construct PRM**
    1. sampling **(nodes)**: **uniformly** sample **n milestones** in the config-space & **collision test**
       a. in free space: kept
       b. collision with obstacle: try again
    2. connection **(edges)**: connect milestones through **linear local path**. **Collision test** along the linear local path by **dense sampling**
       a. if no collision from each point: connect 2 nodes
       b. collision:

▼ **Online: Query**

1. given start/goal point, **connect** them to the **nearest milestone**(node) in PRM by **random walk**.
2. **graph search**: find a path based on cost function (shortest path, least node, etc)
3. termination:
    a. if path returned: always correct
    b. no path returned: error, but hope correct with high probability

- Problem: small connected components, **low connectivity of the whole graph**

▼ **Solutions to increase connectivity:**

- increase #samples

▼ **random walk**

- collision during linear connection: walk in **random direction**
- when hits an obstalce, random walk again **up to a maximum distance** d_{max} —> create **new milestone**.
- try to connect. repeat until 2 subgraphs connect.

—> edges can be no longer straight line, still linear

—> might get you **through narrow passages /opening**

—> resulting path could be very zig-zag: **smoothing necessary!!!**

▼ **path correction**

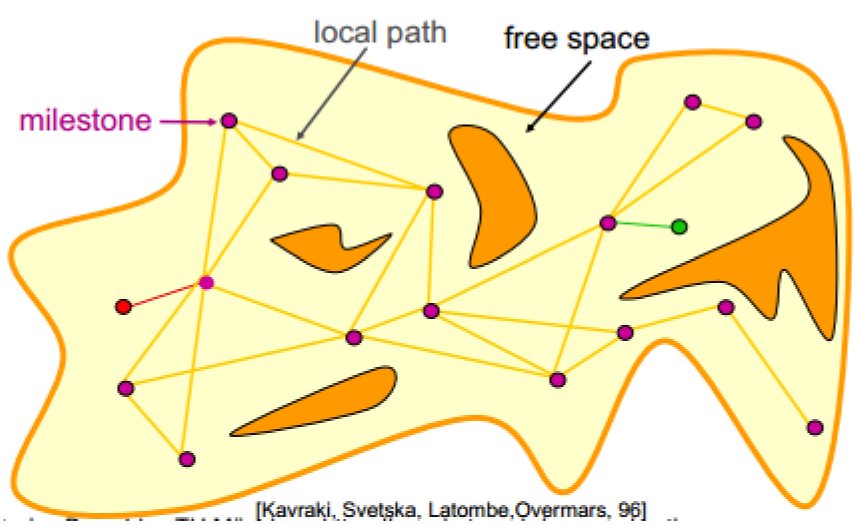
- collision when connecting 2 nodes: push the line towards free space —> **find boundary**
- sample points along the line: for each point, extend **in random direction**
- **binary search:** collision test in by halving the distance **until boundary** —>**create new milestone.**
- connect the milestones with subgraphs.

▼ **sample around obstacles: OBPRM**

- **Smoothing** of zig-zag paths
  - motivation:
    - new nodes that increase connectivity create **zig-zag path,** chaotic & complicated
    - frequent **change of configuration and velocity (negative to positive)**

    —> **smoothing necessary!!!!**
  - Process
    1. connect start to goal — **supporting line**
    2. if collision, find the **most distant node** to the line.
    3. from that node, connects to goal — **supporting line**
    4. repeat, find the **most distant node**, until it reaches the goal.

    —> Path: sequence of these most distant nodes.
  - Preservation of **planning topology: homotopic path**
    - after path correction(smoothing), **homotopy** ensures the **space topology** gets visited from the **same side as planned.**
    - it **doesn't intersect any obstacles**. no jumpting through an obstacle.



Probabilistic Roadmap (PRM) — Multiple-query sampling solving single-query problems

- **single query problem**:
  - **known start & goal**, not far away from each other.
  - **one-time** task
- multiple-query sampling solution:
  - add start & goal with all other uniformly sampled distributed nodes
  - build edges

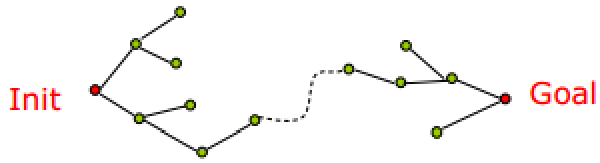
  —> sampling the whole free space uniformly doesn't make sense! —> resource waste

  —> **single-query sampling!!**

Single-query sampling solving single-query problems

▼ Process:

1. **expansion:**
   a. simultaneously **grow 2 trees from start & goal** configuration
   b. **randomly sample n nodes** around existing nodes —> **weighted sampling**
2. **connection:** check if **distance** of child nodes between 2 trees is **below threshold** —> **linear dense sampling** —> **connect** if no collision
3. iterates between expansion and connection until
   a. 2 trees connected
   b. max. number of expansion/connection steps reached



Expansion + Connection

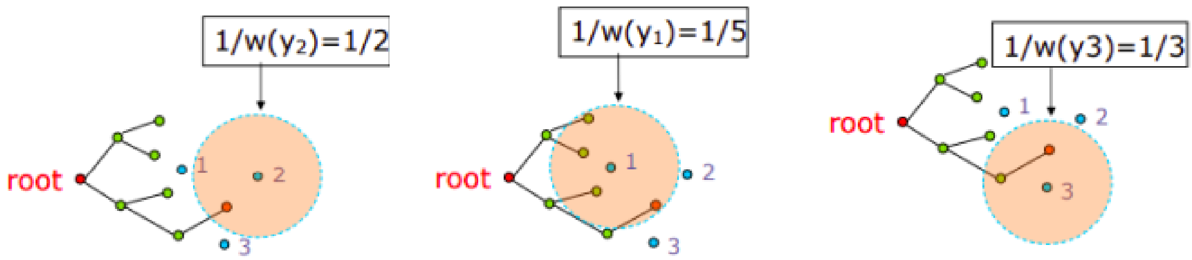
- Problem at sampling:
  - child nodes can **expand back** to the parent nodes (back to known areas)
  - if **obstacle**, the tree will **bounce back** to parent nodes.

    —> unable to go through narrow passages.

- Solution at **Expansion**: **weighted sampling of nodes**
  - Weight: $w(x)$ — **number of existing nodes** (including itself) within **radius d**

  1. pick a node x with probability $\frac{1}{w(x)}$
  2. randomly sample k points around x within radius d
  3. for each sample y, calculate $w(y)$ and **sampling probability** $\frac{1}{w(y)}$
  4. add y into the tree, if
     a. y has higher probability
     b. collision free from y to x (linear collision test)
     c. y can see x

  —> nodes **well distributed** over unknown spaces, also able to **expand to obstacle** or through **narrow passages!!!**



▼ applications:

  - remove parts from car: a collision free path
  - piano mover
  - start & goal close to each other, one-time tasks

Evaluation of PRM: $(\epsilon, \alpha, \beta)$ - Expansiveness

- **coverage**: milestones are distributed such that **almost any point** of config-space can be **connected to one milestone** with **a straight line.** (doesn't have to be connected)
- **connectivity**: a **one-to-one correspondence** between the **connected components** of the roadmap and those of F

  —> difficult in narrow passages

▼ **expansiveness:** characterize both **coverage and connectivity**

  - **visibility set** of q: all configurations seen by q by straight line.
    - **minimum visibility** ϵ**: each configuration** sees **at least** ϵ **fraction** of the **free space**

      —> find the configuration q with minimum visibility

$$\epsilon = \frac{S}{F_{freespace}} \in (0, 1]$$

- β**-lookout set of S** — B, subset of the confined space S: the **area from the confined space S**(minimum visibility) where it **can see the rest of the free space**
- **lookout of B —** β**: each configuration** in B can see **at least** β **fraction** of **freespace** $F \setminus S$

$$\beta = \frac{C}{F \setminus S} \in (0, 1]$$

- α: the fraction of B to the confined space S

$$\alpha = \frac{B}{S}$$

- free space F is $(\epsilon, \alpha, \beta)$**- expansive:**
  - F is ϵ**-good**,
  - for each subset S in F, its β**-lookout is at least** α **fraction of S.**

  —> $\epsilon, \alpha, \beta \uparrow$, construction cost $\downarrow$ for good connectivity and coverage.

  —> **high probability** in getting a **linking sequence** that covers a **large fraction of free space**

Theorem: a roadmap of $n = \frac{8\ln(8/\epsilon \cdot \alpha \cdot \gamma)}{\epsilon \cdot \alpha} + \frac{3}{\beta}$ milestones has the correct connectivity with **at least** $1 - \gamma$ **success rate.**

—> exponential dependency of success and n: **increase n linearly** will result in **exponential improvement of**

- **success**
- **connectivity**
- **coverage**

- basic PRM: uniform random distribution
- not fast for single query
- single query sampling:
  - EST
  - RRT