Optimal Rapidly-exploring Random Trees

CIMAT

Miguel Vargas

Material taken form:

S. Karaman, E. Frazzoli, *Sampling-based Algorithms for Optimal Motion Planning*.

S. Karaman, E. Frazzoli, *Incremental Sampling-based Algorithms for Optimal Motion Planning*.

Introduction

The optimality problem of path planning is to find a feasible path with minimum cost (given a cost function). If no such path exists, report failure.

- The paper provides a systematic and thorough analysis of optimality and complexity for sampling-based path planning algorithms, like Probabilistic RoadMaps (PRM) and Rapidly-exploring Random Trees (RRT). It is proven that PRM and RRT algorithms are not asymptotically optimal.

- New algorithms are proposed PRM*, RRG, and RRT*. These are proven to be probabilistycally complete, asymptotically optimal and computationally efficient.

- The key insight is that connections between vertices in the graph should be sought within balls whose radius vanishes with a certain rate as the size of the graph increases, and is based on new connections between motion planning and thoery of random geometric graphs.

In this work we will discuss only RRT and RRT* algorithms.

Notation

Let $X = (0,1)^d$ be the configuration space of dimension d. Let X_{obs} be the obstacle region, and $X_{\text{free}} = \text{cl}\left(X \setminus X_{\text{obs}}\right)$ the obstacle-free region, where $\text{cl}(\cdot)$ is the closure of a set.

The initial condition $x_{\text{init}} \in X_{\text{free}}$. The goal region X_{goal} is an open subset of X_{free}.

Given a set $X \subset \mathbb{R}^d$, and a scalar $s \geq 0$, a path in X is a continuous function

$$\sigma : [0, s] \rightarrow X,$$

where s is the length of the path.

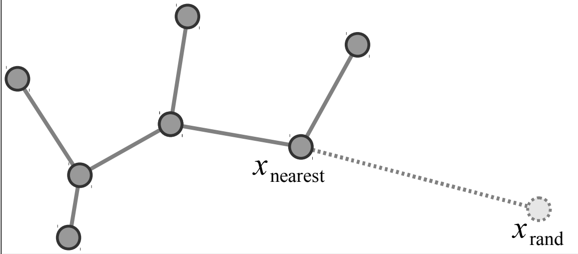
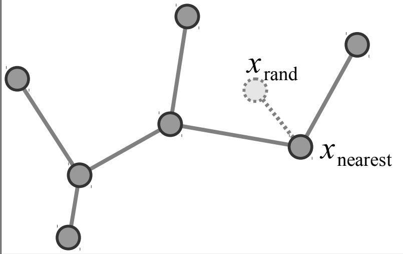
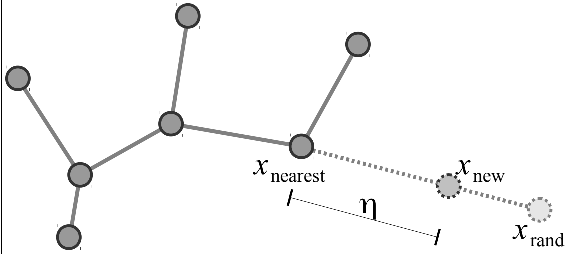
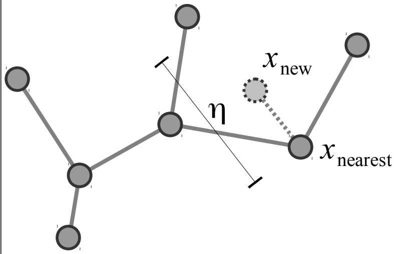
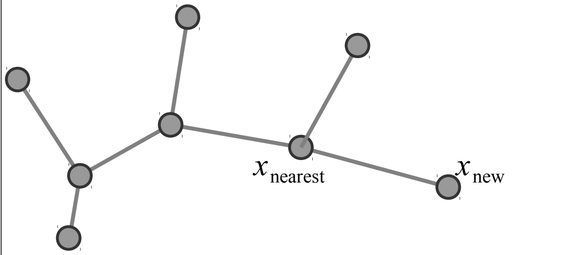
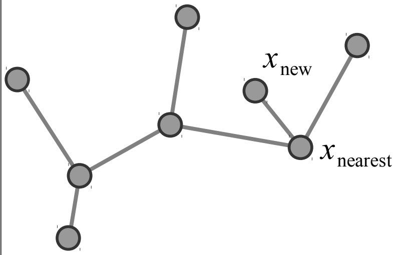
Given two paths in X, $\sigma_1 : [0, s_1] \rightarrow X$ and $\sigma_2 : [0, s_2] \rightarrow X$, with $\sigma_1(s_1) = \sigma_2(0)$, their concatenation is denoted $\sigma_1 | \sigma_2$,

$$\sigma = \sigma_1 | \sigma_2 : [0, s_1 + s_2] \rightarrow X.$$

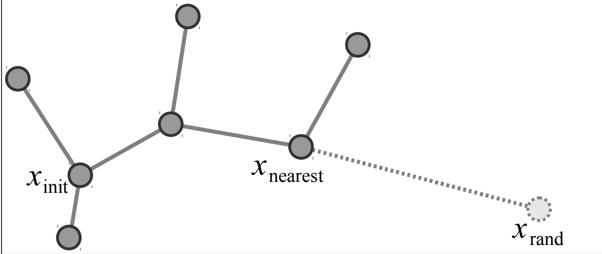
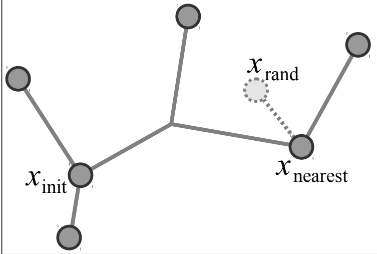
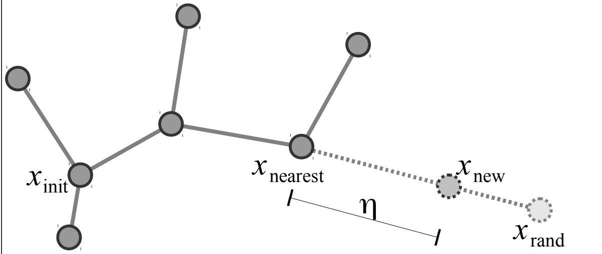
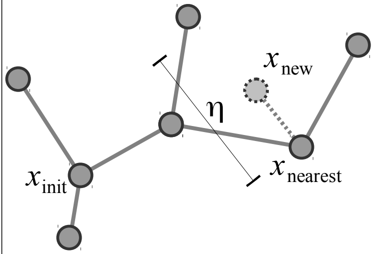
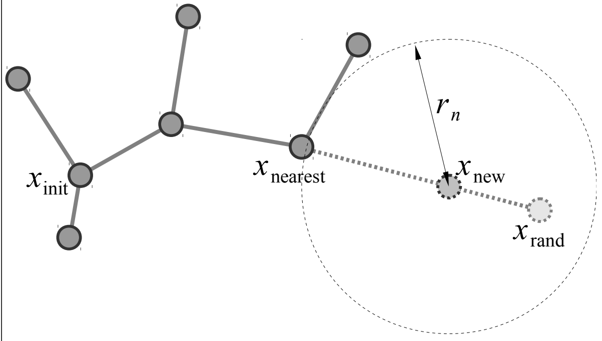
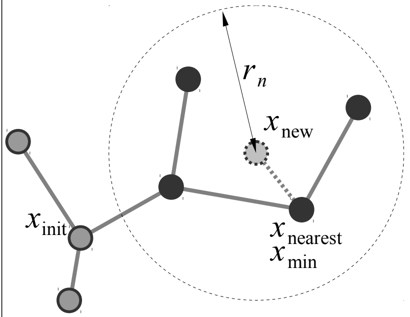
Ths set of all paths in X with nonzero length is denoted by Σ.

The closed Ball of radius $r > 0$ centered at $x \in \mathbb{R}^d$ is defined as $B_{x,r} := \left\{ y \in \mathbb{R}^d \,\middle|\, \|y - x\| \leq r \right\}$.

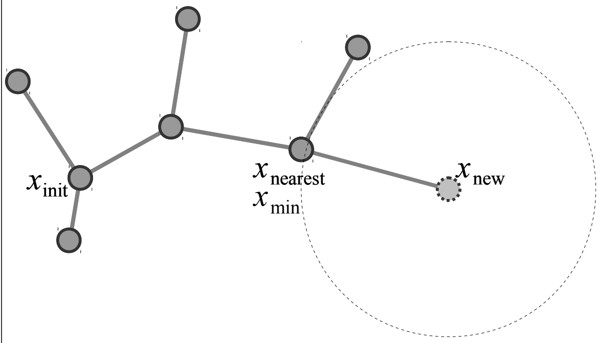
RRT

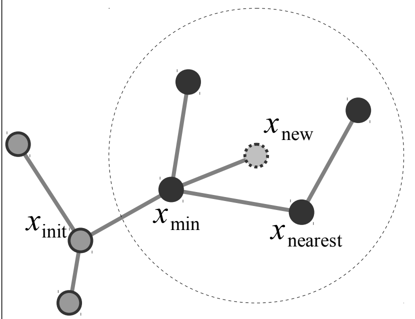
Algorithm	Example 1	Example 2
$V \leftarrow \{x_{\text{init}}\}$ $E \leftarrow \emptyset$ **for** $i \leftarrow 1 \ldots N$		
$x_{\text{rand}} \leftarrow \text{SampleFree}(i)$ $x_{\text{nearest}} \leftarrow \text{Nearest}((V, E), x_{\text{rand}})$		
$x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x_{\text{rand}})$	 x_{nearest} x_{new} η x_{rand}	
if $\text{ObstacleFree}(x_{\text{nearest}}, x_{\text{new}})$ $V \leftarrow V \cup \{x_{\text{new}}\}$ $E \leftarrow E \cup \{(x_{\text{nearest}}, x_{\text{new}})\}$		
return (V, E)		

RRT*

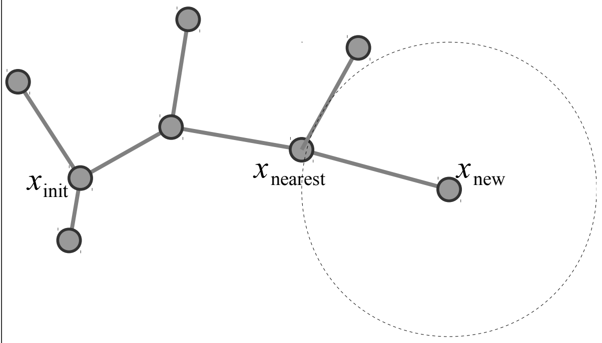
Algorithm	Example 1	Example 2
$V \leftarrow \{x_{\text{init}}\}$ $E \leftarrow \emptyset$ **for** $i \leftarrow 1 \ldots N$		
$x_{\text{rand}} \leftarrow \text{SampleFree}(i)$ $x_{\text{nearest}} \leftarrow \text{Nearest}((V, E), x_{\text{rand}})$	 x_{init} x_{nearest} x_{rand}	 x_{rand} x_{init} x_{nearest}
$x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x_{\text{rand}})$	 x_{init} x_{nearest} x_{new} η x_{rand}	 x_{new} η x_{init} x_{nearest}
if $\text{ObstacleFree}(x_{\text{nearest}}, x_{\text{new}})$ $X_{\text{near}} \leftarrow \text{Near}((V, E), x_{\text{new}}, r_n)$ $V \leftarrow V \cup \{x_{\text{new}}\}$ $x_{\text{min}} \leftarrow x_{\text{nearest}}$ $c_{\text{min}} \leftarrow \text{Cost}(x_{\text{nearest}}) + \text{c}(\text{Line}(x_{\text{nearest}}, x_{\text{new}}))$	 r_n x_{init} x_{nearest} x_{new} x_{rand}	 r_n x_{new} x_{init} x_{nearest} x_{min}

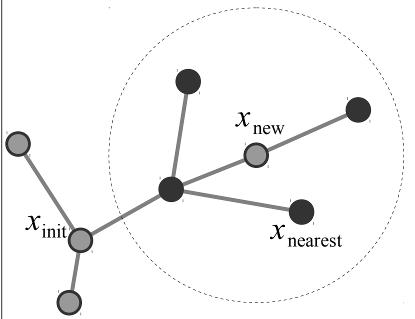
```
**for_each** $x_{\text{near}} \in X_{\text{near}}$
  **if** $\text{CollisionFree}(x_{\text{near}}, x_{\text{new}})$
    **if** $\text{Cost}(x_{\text{near}}) + \text{c}(\text{Line}(x_{\text{near}}, x_{\text{new}})) < c_{\text{min}}$
      $x_{\text{min}} \leftarrow x_{\text{near}}$
      $c_{\text{min}} \leftarrow \text{Cost}(x_{\text{near}}) + \text{c}(\text{Line}(x_{\text{near}}, x_{\text{new}}))$
$E \leftarrow E \cup \{(x_{\text{min}}, x_{\text{new}})\}$
```



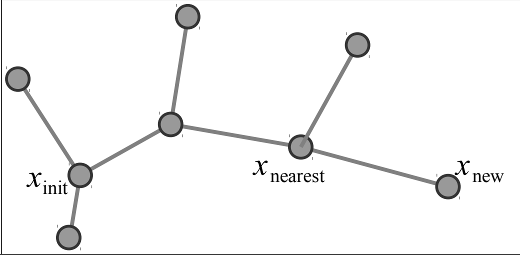


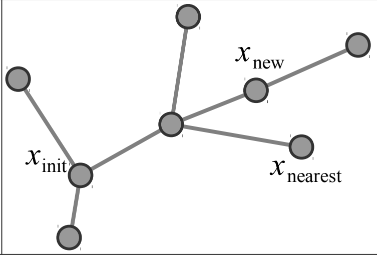
```
**for_each** $x_{\text{near}} \in X_{\text{near}}$
  **if** $\text{CollisionFree}(x_{\text{new}}, x_{\text{near}})$
    $t \leftarrow \text{Cost}(x_{\text{new}}) + \text{c}(\text{Line}(x_{\text{new}}, x_{\text{near}}))$
    **if** $t < \text{Cost}(x_{\text{near}})$
      $x_{\text{parent}} \leftarrow \text{Parent}(x_{\text{near}})$
      $E \leftarrow (E \setminus \{(x_{\text{parent}}, x_{\text{near}})\}) \cup \{(x_{\text{new}}, x_{\text{near}})\}$
```





```
**return** $(V, E)$
```





The RRT* algorithm essentially "rewires" the tree as it discovers new lower-cost paths reaching the nodes that are already in the tree.

Near function

This function returns a set with all nodes of the tree within a ball of radius r_n centered in x_{new}.

Ball radius.
This radius is determined with

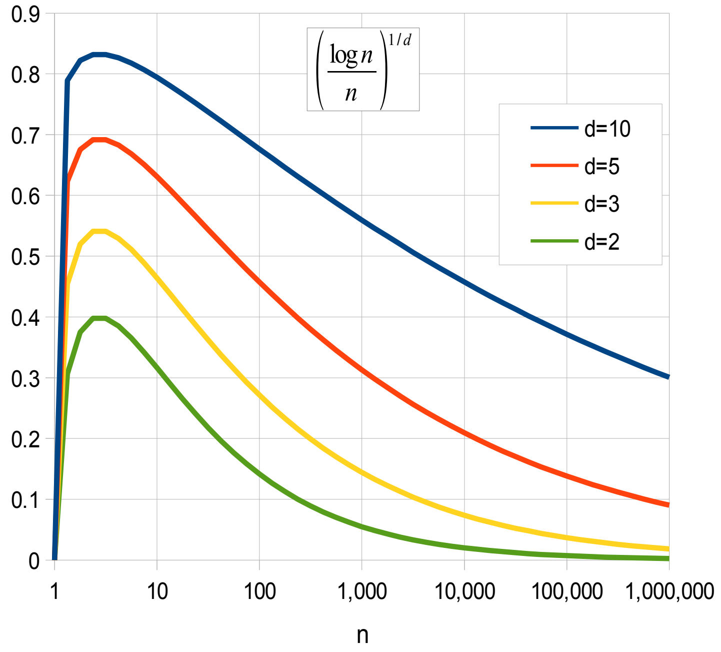
$$r_n \leftarrow \min\left\{\gamma\left(\frac{\log n}{n}\right)^{1/d}, \eta\right\},$$

with

$$\gamma \leftarrow 2\left(1+\frac{1}{d}\right)^{1/d}\left(\frac{\mu\left(X_{\text{free}}\right)}{\zeta_d}\right)^{1/d},$$

where

$$n \leftarrow |V|,$$



is the cardinality of the set of nodes, $\mu\left(X_{\text{free}}\right)$ denotes the Lebesgue measure (i.e., volume) of the obstacle-free space, and ζ_d the volume of the unit ball in the d-dimensional Euclidean space.

The dispersion of a set of n random points sampled uniformly and independently in a bounded set $S \subset \mathbb{R}^d$ is $O\left(\left(\frac{\log n}{n}\right)^{1/d}\right)$. Dispersion captures the degree to which points in a point set are separated from each other.

Cost function

It returns a distance from a node of the tree to the root, let $v \in V$,

$$\mathrm{Cost}(v) = \mathrm{Cost}(\mathrm{Parent}(v)) + c(\mathrm{Line}(\mathrm{Parent}(v), v)).$$

If $v_0 \in V$ is the root of the tree, then $\mathrm{Cost}(v_0) = 0$.

c function

Let $c : \Sigma \rightarrow R_{>0}$ be a function, called cost function, which assigns a non-negative cost to all nontrivial collision-free paths.

Line function

Given two points $x_1, x_2 \in \mathbb{R}^d$, the function

$$\mathrm{Line}(x_1, x_2) : [0, s] \rightarrow X$$

is defined as the straight line path from x_1 to x_2.

Probabilistic completeness

For any robustly feasible path planning problem $(X_{\text{free}}, x_{\text{init}}, X_{\text{goal}})$, there exist a constants $a>0$ and $n_0 \in \mathbb{N}$, both dependent only on X_{free} and X_{goal}, such that

$$P\left(\left\{V_n^{\text{RRT}} \cap X_{\text{goal}} \neq \emptyset\right\}\right) > 1 - \frac{1}{e^{a n}}, \ \forall n > n_0;$$

also

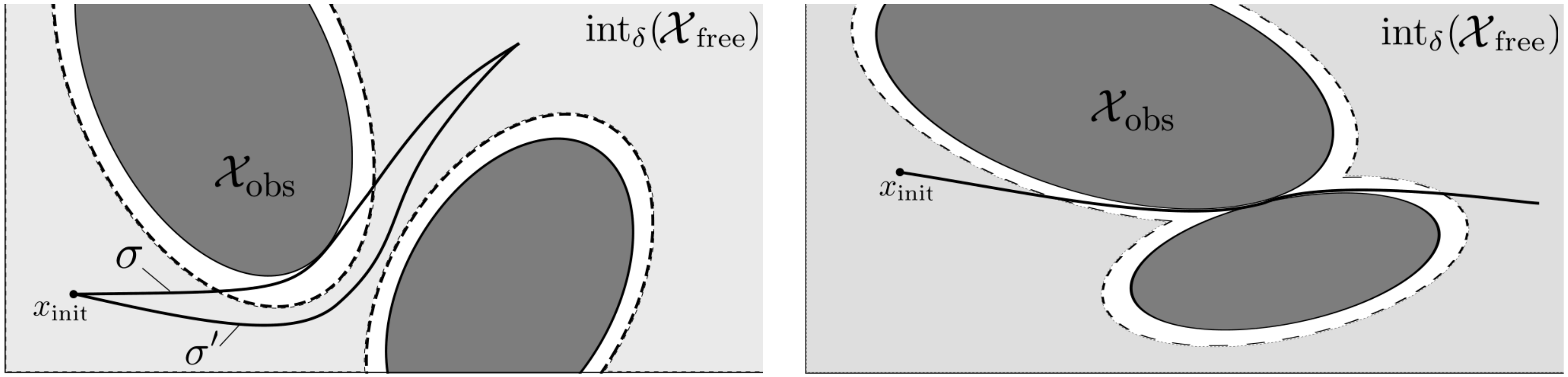
$$P\left(\left\{V_n^{\text{RRT*}} \cap X_{\text{goal}} \neq \emptyset\right\}\right) > 1 - \frac{1}{e^{a n}}, \ \forall n > n_0.$$

Proof.
By construction $V_n^{\text{RRT*}}(\omega) = V_n^{\text{RRT}}(\omega)$, for all $\omega \in \Omega$, and $n \subset \mathbb{N}$. RRT* returns a connected graph. Hence the result follows directly from the probabilistic completeness of RRT. □

If the RRT algorithm returns a feasible solution by iteration n, so will the RRT* algorithm, assuming the same sample sequence.

Asymptotically optimality

A collision-free path $\sigma:[0,s]\rightarrow X$ is said to have weak $\delta-$clearance, if exists a path σ' that has strong $\delta-$clearance, and there exist a homotopy between them.



$\mathrm{int}_\delta(\mathcal{X}_{\mathrm{free}})$
$\mathcal{X}_{\mathrm{obs}}$
σ
x_{init}
σ'
$\mathrm{int}_\delta(\mathcal{X}_{\mathrm{free}})$
$\mathcal{X}_{\mathrm{obs}}$
x_{init}

A feasible path $\sigma^*\in X_{\mathrm{free}}$ that solves the optimaly problem is said to be robustly optimal if it has a weak $\delta-$clearance and, for any sequence of collision free paths $\{\sigma_n\}_{n\in\mathbb{N}}$, $\sigma_n\in X_{\mathrm{free}}\,\forall\, n\in\mathbb{N}$, such that $\lim_{n\rightarrow\infty}\sigma_n=\sigma*$, $\lim_{n\rightarrow\infty}c(\sigma_n)=c(\sigma*)$.

Let $c^*=c(\sigma*)$ be the cost of an optimal path, and let Y_n^{RRT} be the extended random variable corresponding to the cost of the minimum-cost solution included in the graph returned by RRT.

An algorithm ALG is asymptotically optimal if, for any path planning problem $(X_{\mathrm{free}}, x_{\mathrm{init}}, X_{\mathrm{goal}})$, and cost function $c:\Sigma\rightarrow\mathbb{R}_{\geq 0}$ that admit a robustly optimal solution with finite cost c^*

$$P\left(\left\{\limsup_{n\rightarrow\infty} Y_n^{\mathrm{RRT}}=c^*\right\}\right)=1.$$

The RRT algorithm is not asymptotically optimal

Each iteration of the RRT algorithm either adds a vertex and an edge, or leages the graph unchanged.

The limit $\lim_{n\to\infty}\sup Y_n^{\mathrm{RRT}}$ exists and is equal to the random variable Y_∞^{RRT}. This limit is strictly greater that c^* almost surely,

$$P\left(\left\{\lim_{n\to\infty}\sup Y_n^{\mathrm{RRT}} > c^*\right\}\right) = 1.$$

The cost of the best solution returned by RRT converges to a suboptimal value, with probability one.

Asymptotic optimality of RRT*

Assumptions:

1. The cost function is additive.
For all $\sigma_{1,}\sigma_2 \in \Sigma_{X_{\text{free}}}$, the cost function c statisfies the following: $c(\sigma_1|\sigma_2) = c(\sigma_1) + c(\sigma_2)$.

2. Continuity of the cost function.
The cost function c is Lipschitz continuous in the following sense: there exists some constant κ such that for any two paths $\sigma_1 : [0, s_1] \rightarrow x_{\text{free}}$ and $\sigma_s : [0, s_2] \rightarrow x_{\text{free}}$,

$$|c(\sigma_1) - c(\sigma_2)| \leq \kappa \sup_{\tau \in [0,1]} \|\sigma_1(\tau_1) - \sigma_2(\tau_2)\|.$$

3. Obstacle spacing.
There exists a constant $\delta \in \mathbb{R}_+$ such that for any point $x \in X_{\text{free}}$ there exits $x' \in X_{\text{free}}$, such that

i. The δ-ball centered at x' lines inside X_{free},

$$B_{x',\delta} \subset X_{\text{free}}.$$

ii. x lies inside the δ-ball centered at x',

$$x \in B_{x',\delta}.$$

The following theorem ensures the asymptotic optimality of the RRT* algorithm.

Theorem 2.
Let y_i denote the cost of the minimum cost path in the tree, at the iteration i.
Taking asumptions 1, 2 and 3. Then, the cost of the minimum cost path in the RRT* converges to c^* almost surely,

$$P\left(\left\{\lim_{i\to\infty} y_i = c^*\right\}\right) = 1$$

Theorem 3.
If

$$\gamma > 2\left(1+\frac{1}{d}\right)^{1/d}\left(\frac{\mu\left(X_{\text{free}}\right)}{\zeta_d}\right)^{1/d},$$

then the RRT* algorithm is asymtotically optimal.

Complexity of RRT vs RRT*

- The number of calls for **Sample**, **Streer** and **Nearest** is the same for both algorithms.

- **ObstacleFree** is called only once in RRT and could be called many times by RRT*.

- RRT* also uses **Near** and **Cost** functions.
  - **Cost** is $O(\log n)$.
  - An optimal **Near** function is $O\left(\log n + (1/\epsilon)^{d-1}\right)$ (Arya, Mount. 2000).

Let N_i the number of vertices at the end of iteration i. Let M_i^{RRT} and $M_i^{\mathrm{RRT*}}$ be the random variable that denotes the number of steps taken by RRT and RRT* algorithm in iteration i.

Assuming that **Nearest** is implemented an algorithm optimal in fixed dimensions (Arya, Mount, 1999), the number of steps executed by the RRT algorithm at each iteration is at least order $\log(N_i)$ in expectation in the limit, i.e., there exists a constant $\phi \in \mathbb{R}_{>0}$ such that

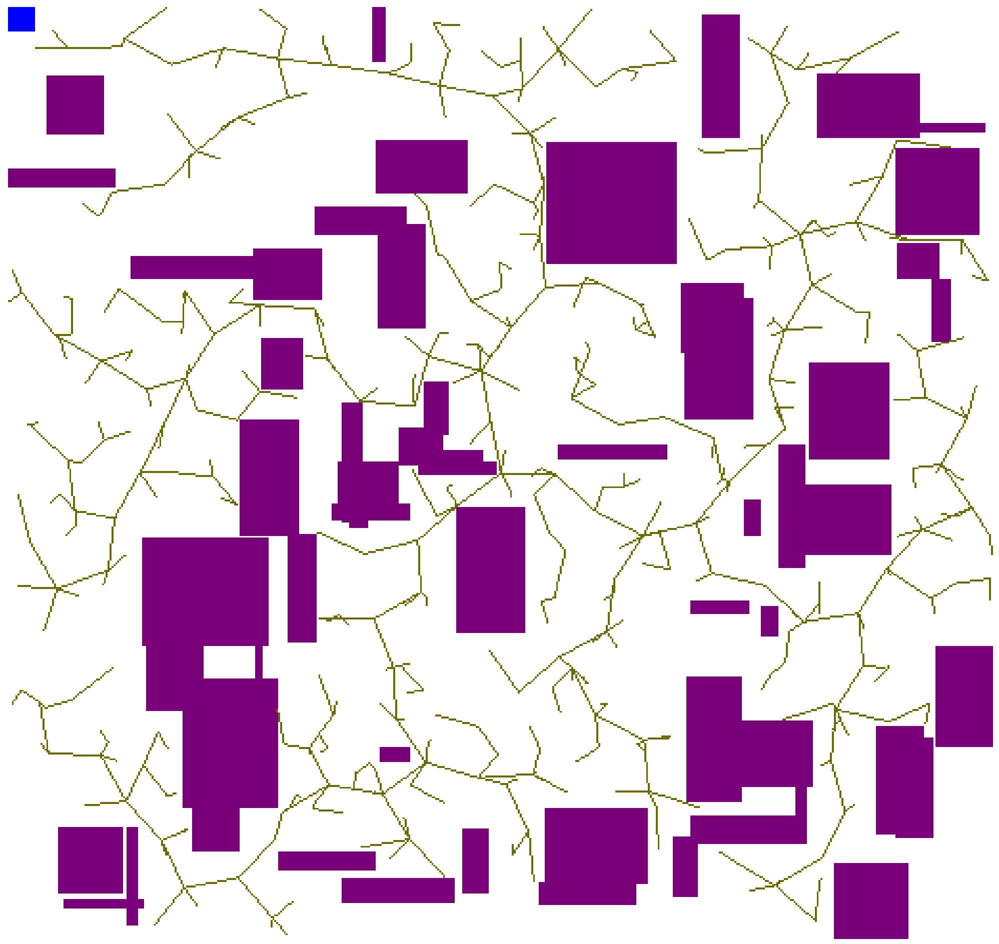
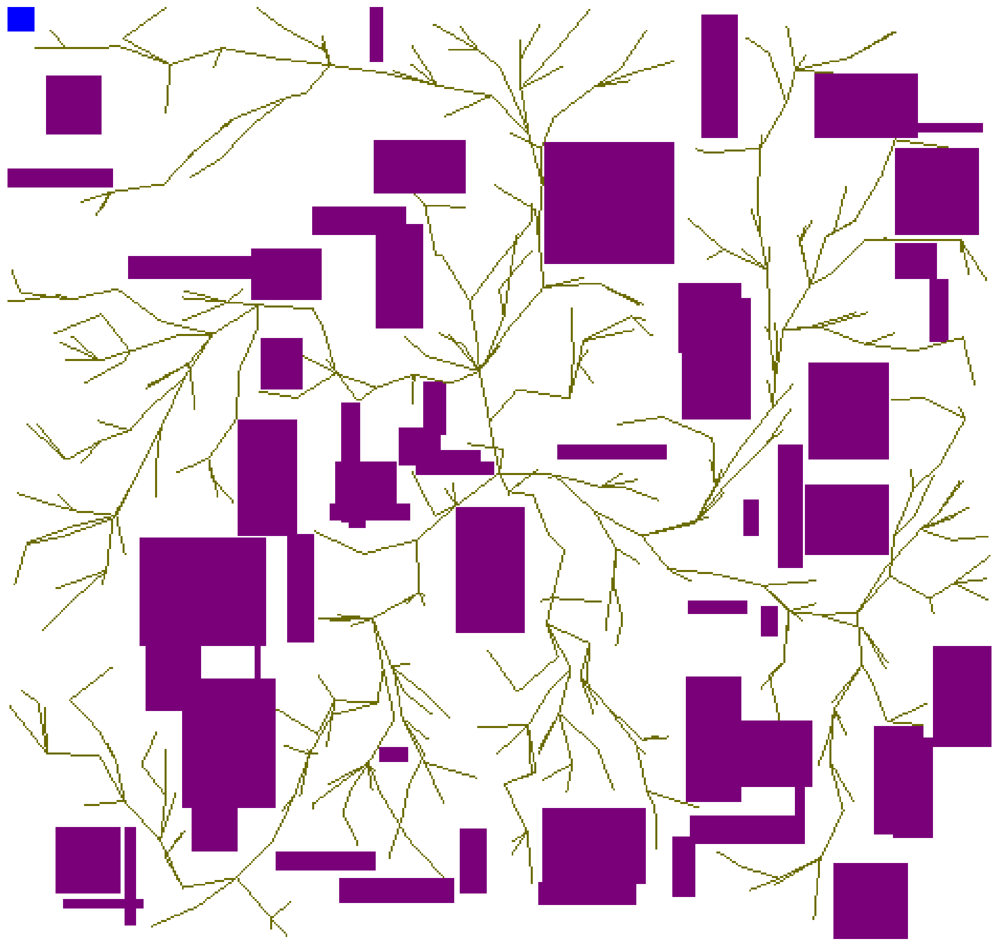
$$\liminf_{i\to\infty} E\left[\frac{M_i^{\mathrm{RRT}}}{\log(N_i)}\right] \geq \phi.$$

Under the previous assumption, there exists a constant $\phi \in \mathbb{R}_{>0}$ such that

$$\limsup_{i\to\infty} E\left[\frac{M_i^{\mathrm{RRT*}}}{M_i^{\mathrm{RRT}}}\right] \leq \phi.$$

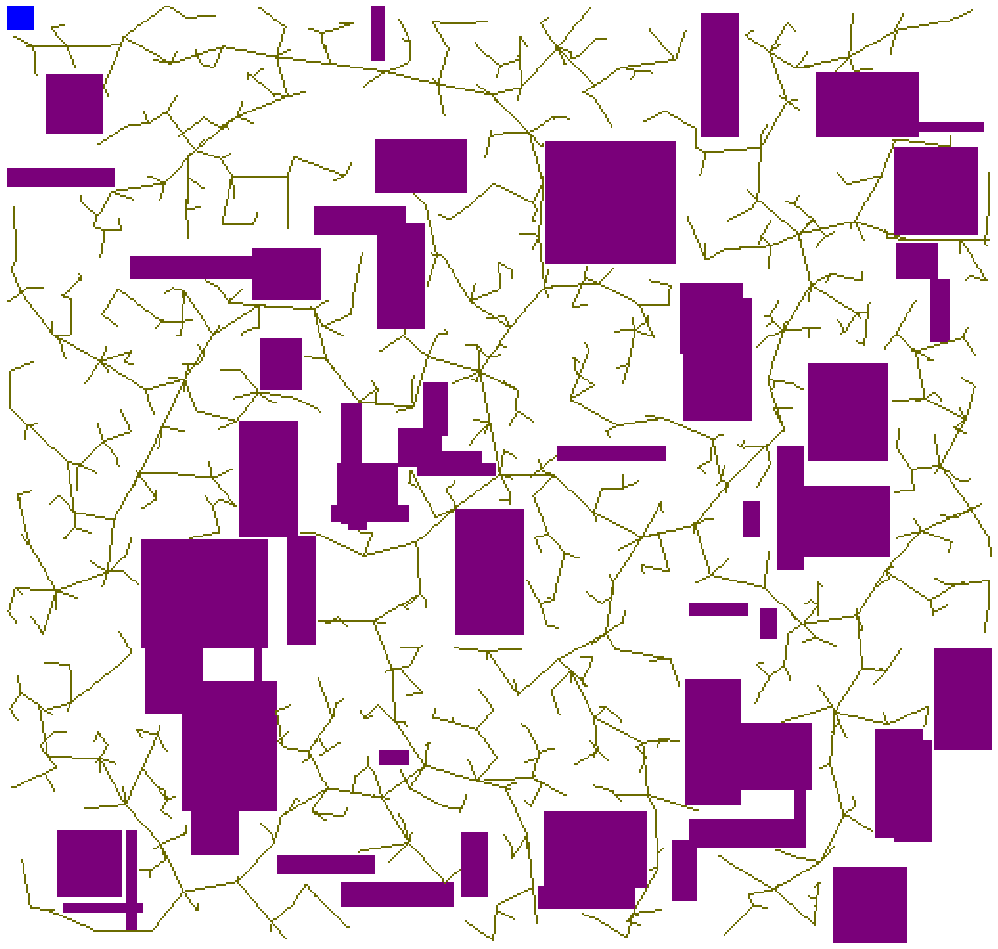
The RRT* algorithm does not have significant overhead when compared to RRT algorithm in terms of asymptotic computational complexity. This is also supported by experimental evidence.

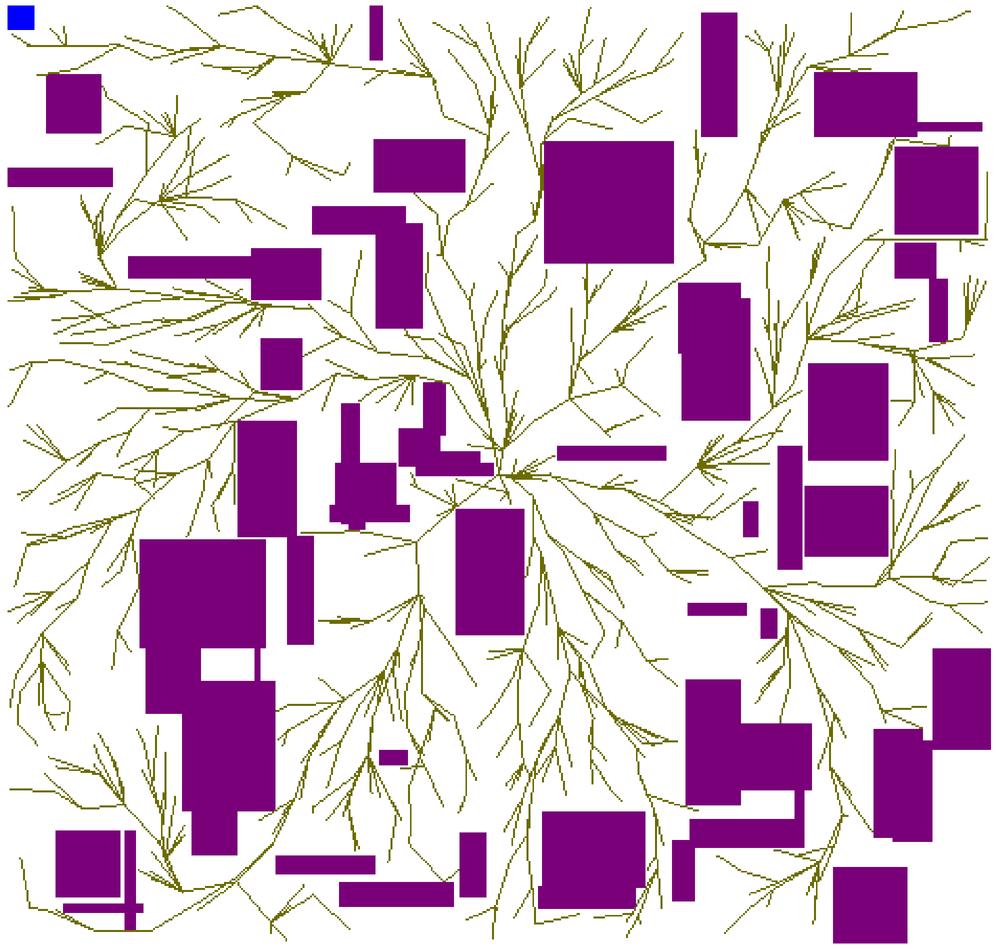
Simulations, RRT vs RRT*

503 iterations

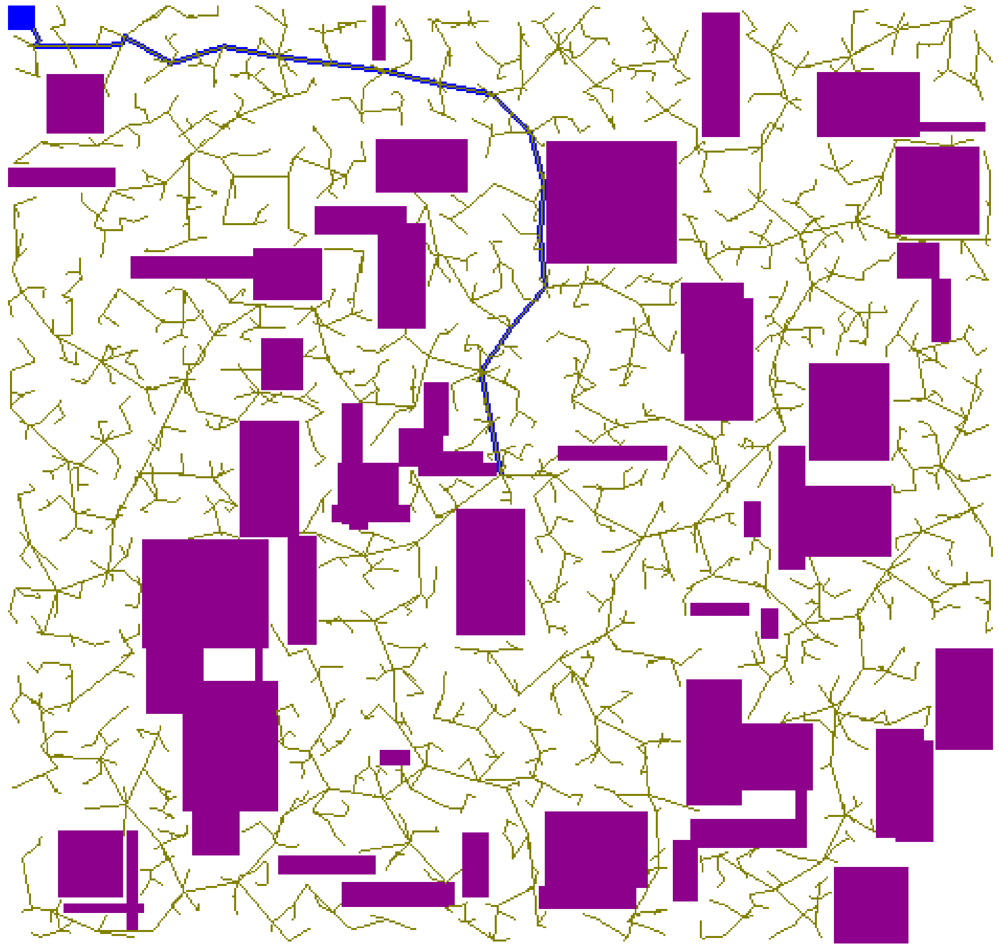
Simulations, RRT vs RRT*

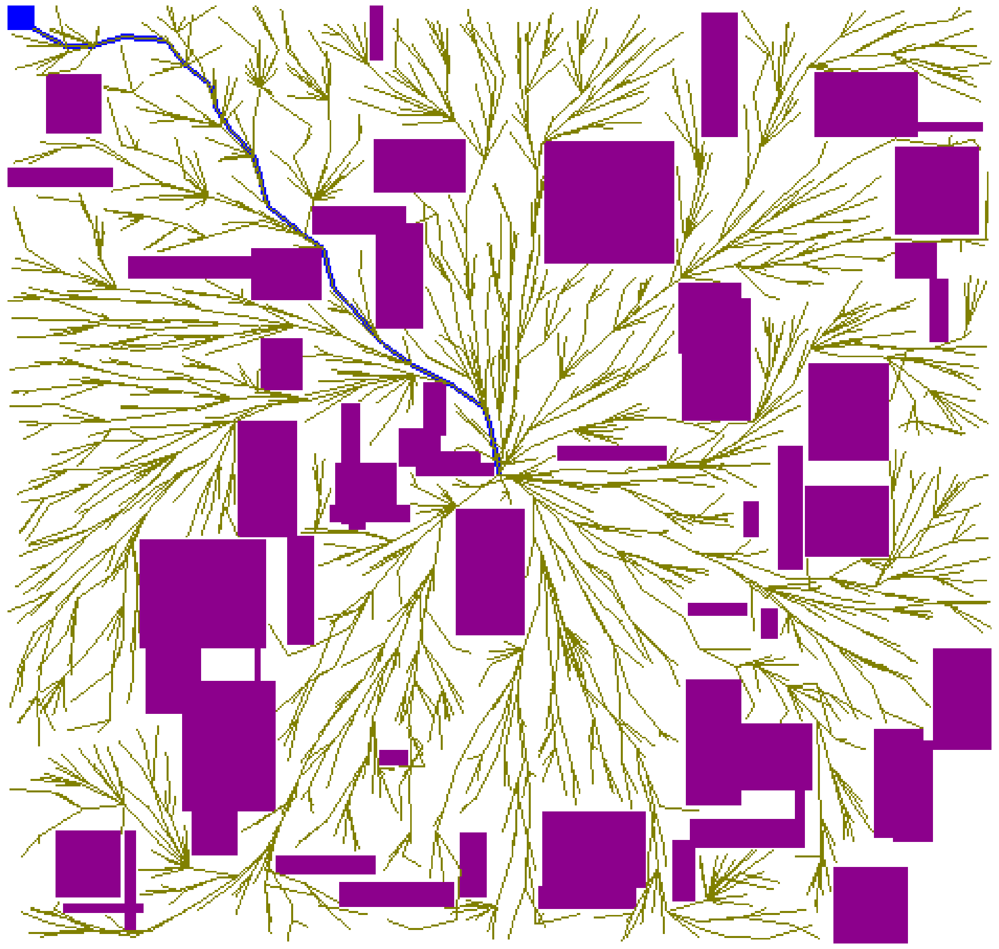




1027

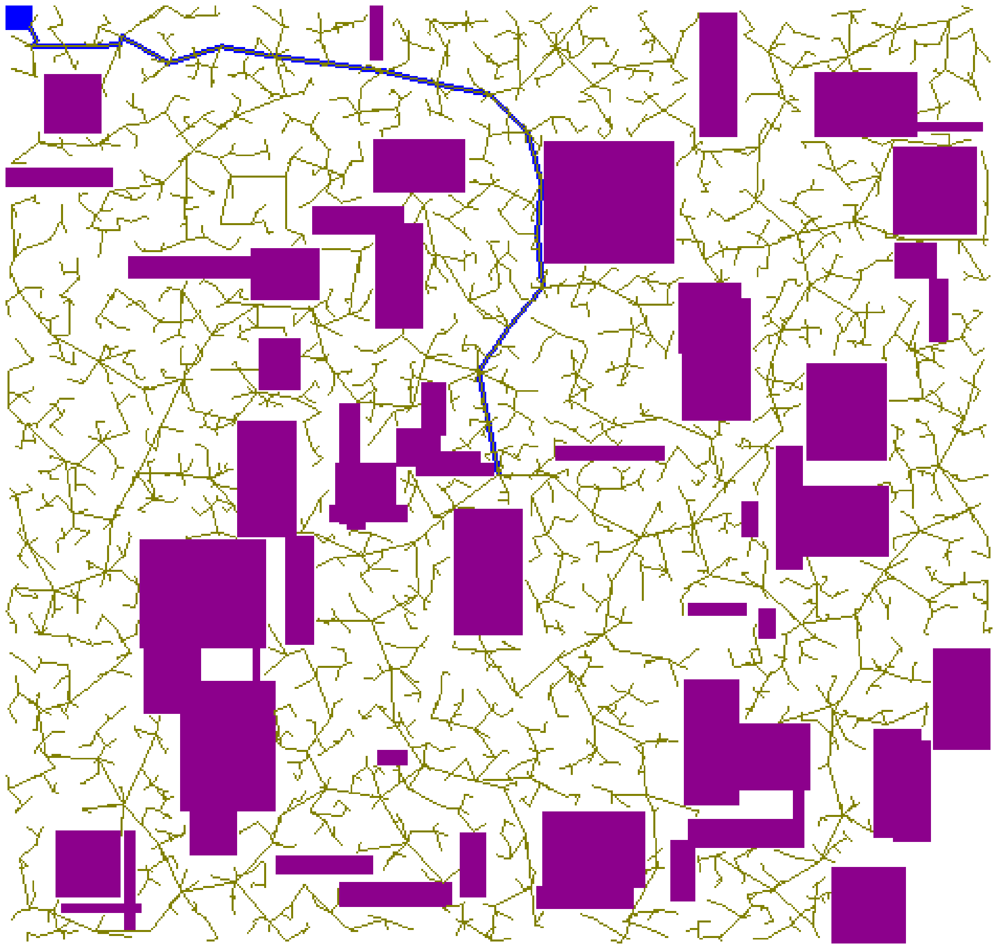
Simulations, RRT vs RRT*

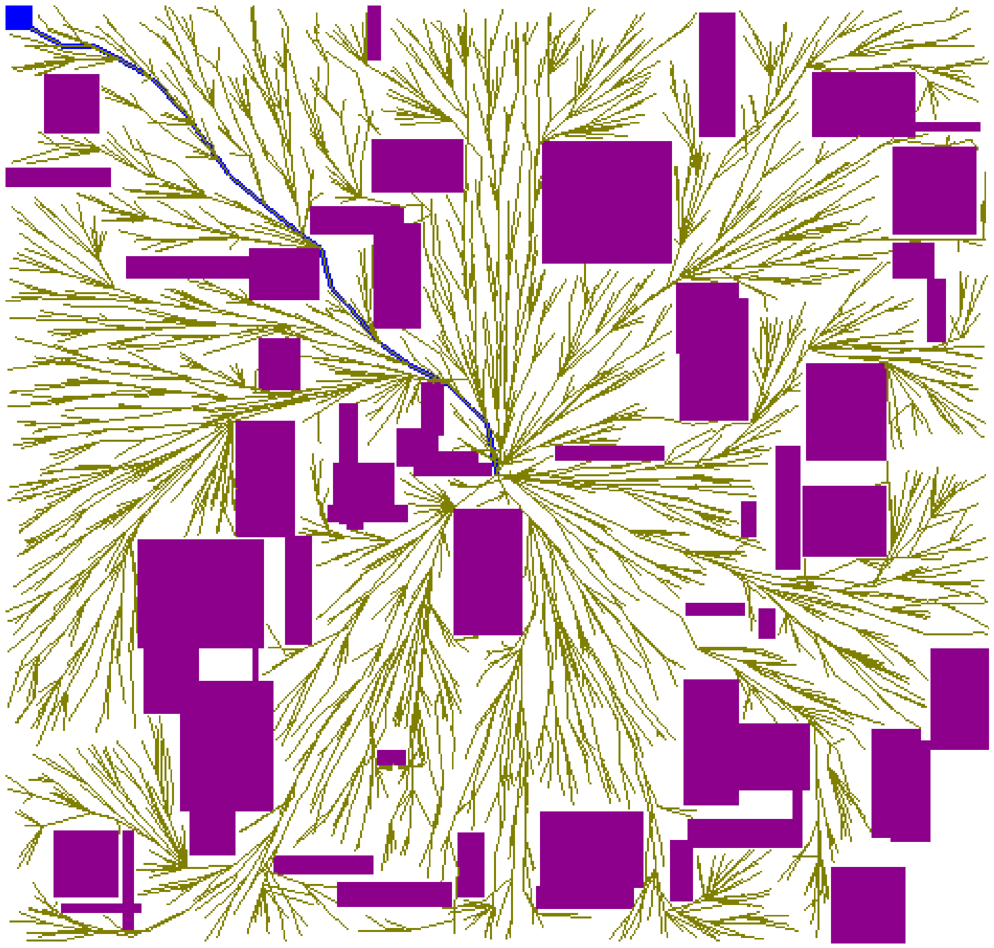




2062

Simulations, RRT vs RRT*



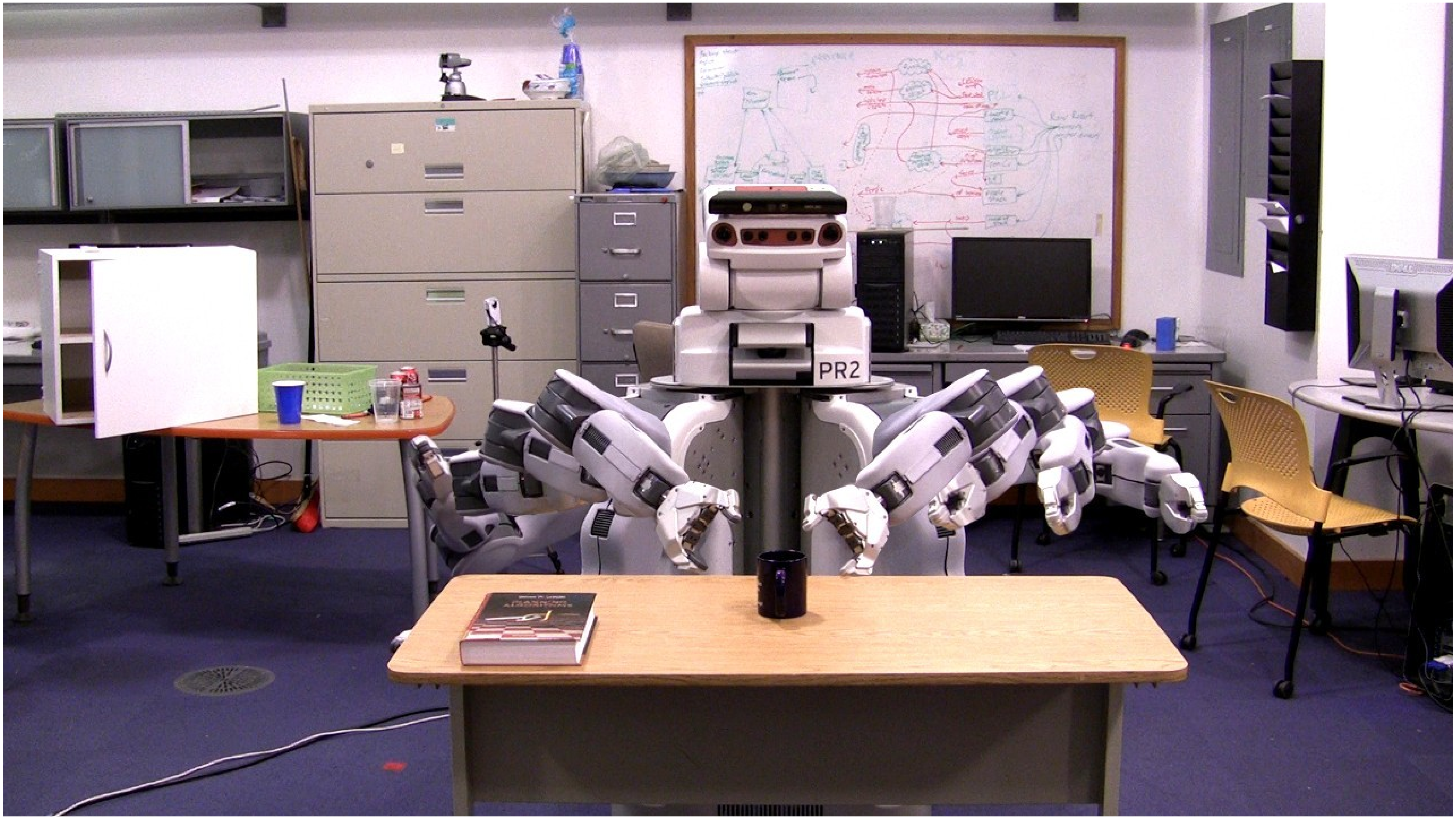


3037

Example RRT

http://www.youtube.com/watch?v=vW74bC-Ygb4

Example RRT*



http://www.youtube.com/watch?v=2WOBMswcCA8

References

S. Karaman, E. Frazzoli, *Sampling-based Algorithms for Optimal Motion Planning*.

S. Karaman, E. Frazzoli, *Incremental Sampling-based Algorithms for Optimal Motion Planning*.

S. Arya, D. M. Mount, R. Silverman, and A. Y. Wu. *An optimal algorithm for approximate nearest neighbor search in fixed dimensions*. Journal of the ACM, 45(6):891–923, November 1999.

S. Arya and D. M. Mount. Approximate range searching. Computational Geometry: Theory and Applications, 17:135–163, 2000.

http://sertac.scripts.mit.edu/rrtstar