RDT (Rapidly growing Dense Tree)
RRT (Rapidly growing Random Tree)

RDT: Idea

- Incrementally construct a search tree
- Gradually improve resultion
- Without setting parameters for resolution

  → Tree **densly** covers C-space
- A dense sequence of samples is used to guide incremental construction of the tree
- Dense sequence = randomly generated → RDT → RRT
- RRT is part of the RDT family

RDT: strategy

- $\alpha(i)$: infinite, dense sequence of samples in C-space
- Possible ways to get a sequence
- Uniform
- Probabilistic (dense with probability one)
- Random with nonuniform bias allowed, as long it is dense with probability one.
- RDT is a topological graph G(V,E)
- $S \subset C_{free}$: all points reached by G

RDT: Algorithm (Simplest one)

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 $q_n \leftarrow$ nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));

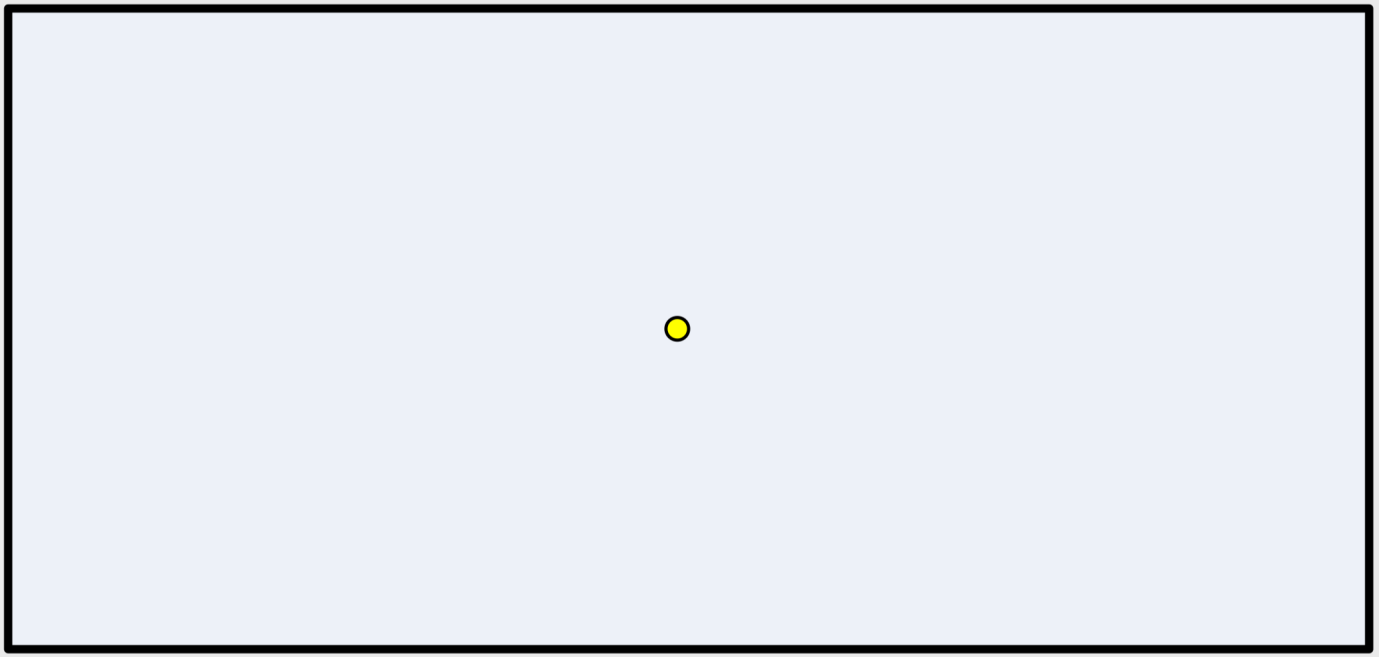
- α(i): infinite, dense sequence
- Above algorithm is the basic one when no obstacles are present.

Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));

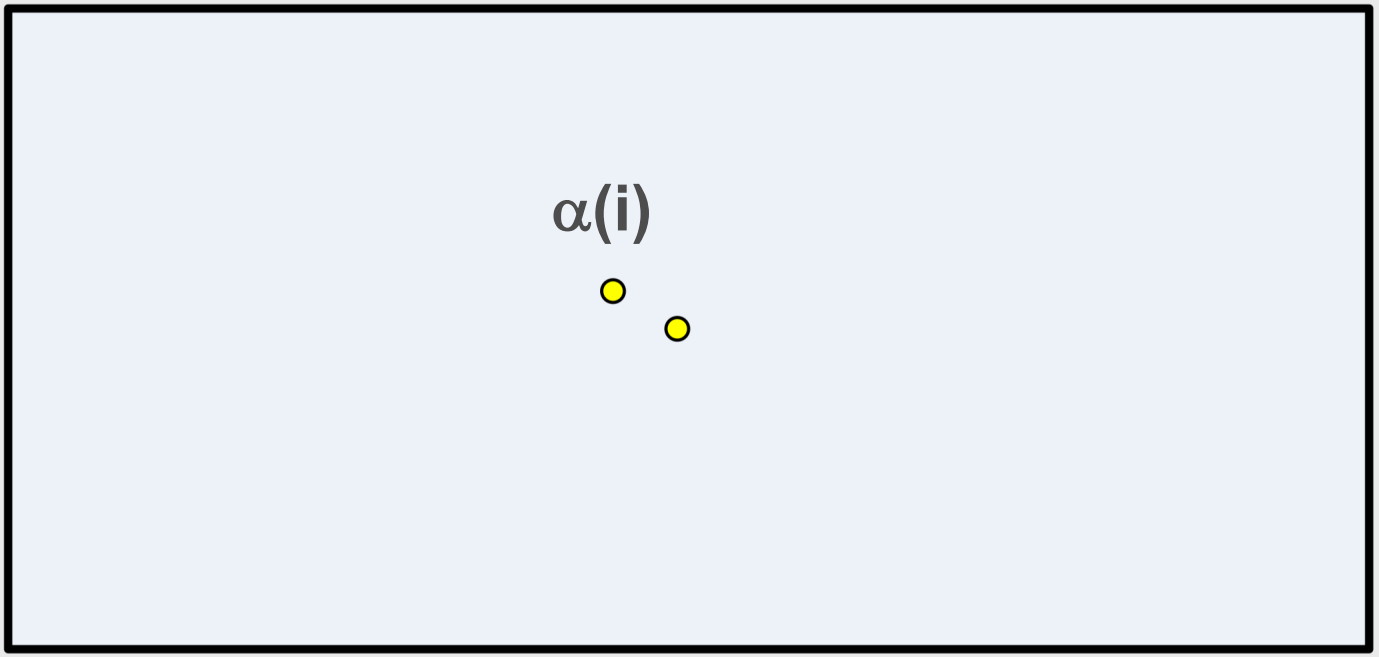
Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));



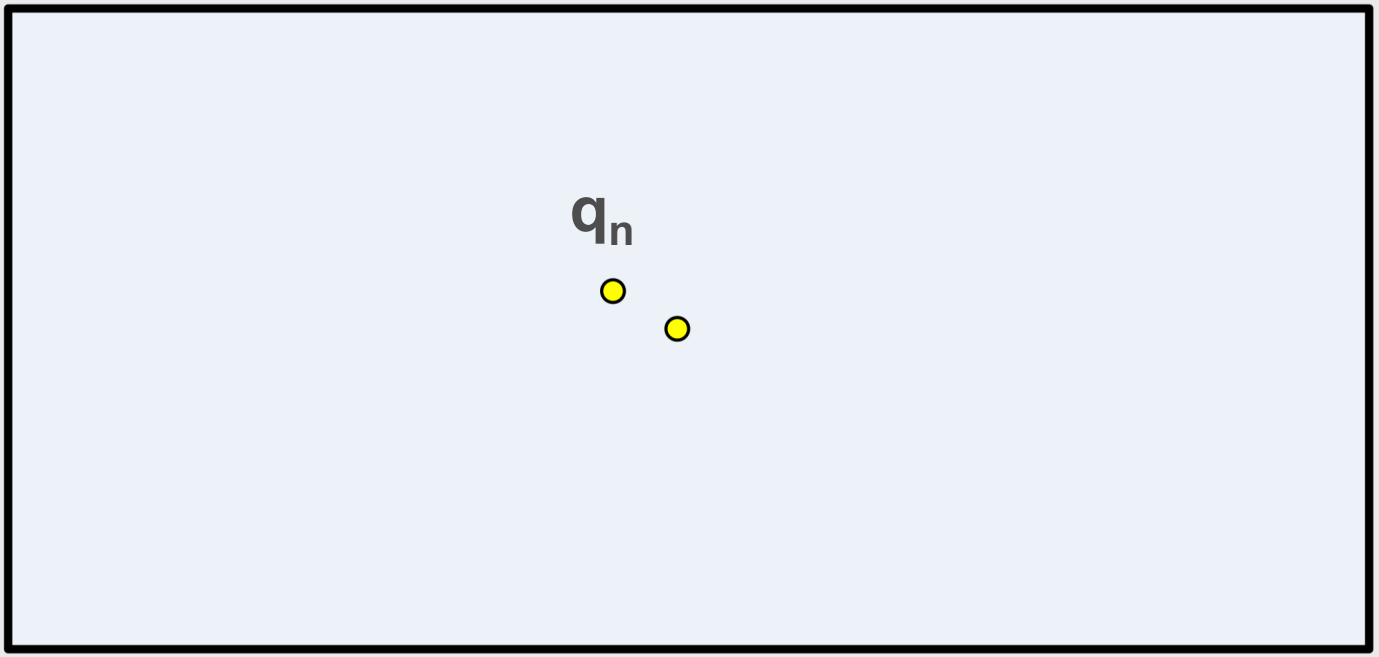
Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));



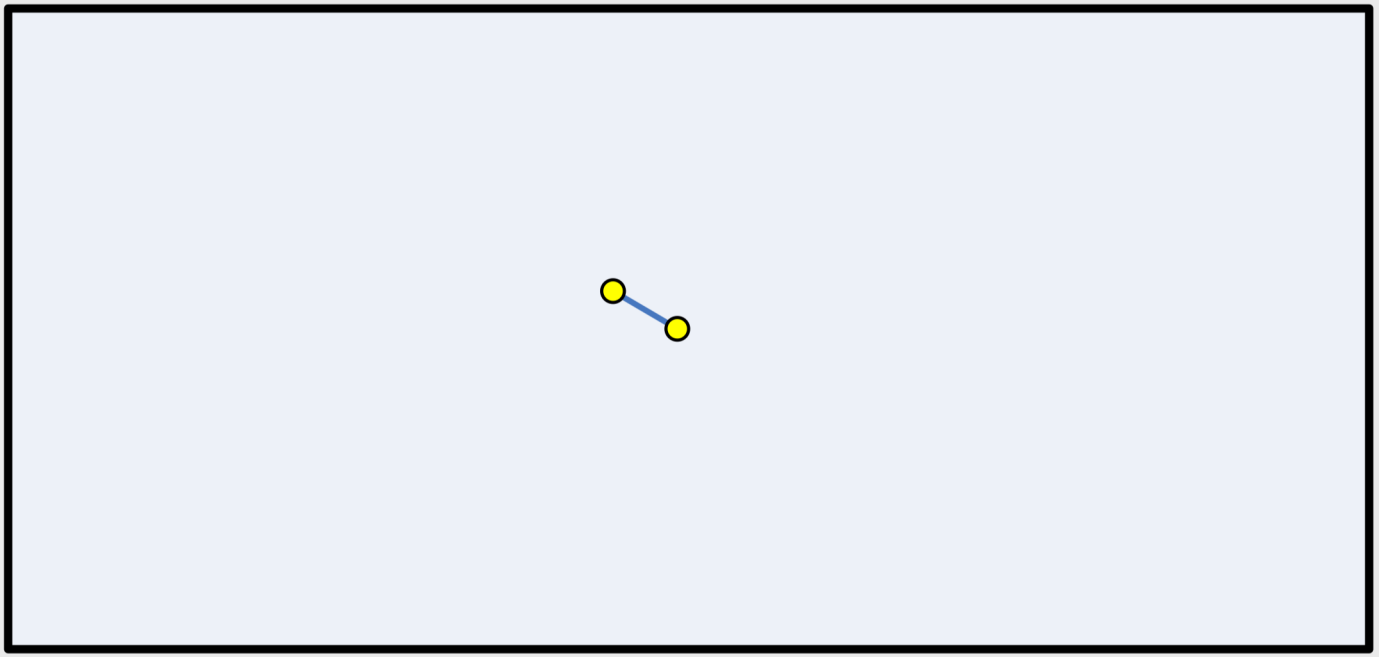
Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));



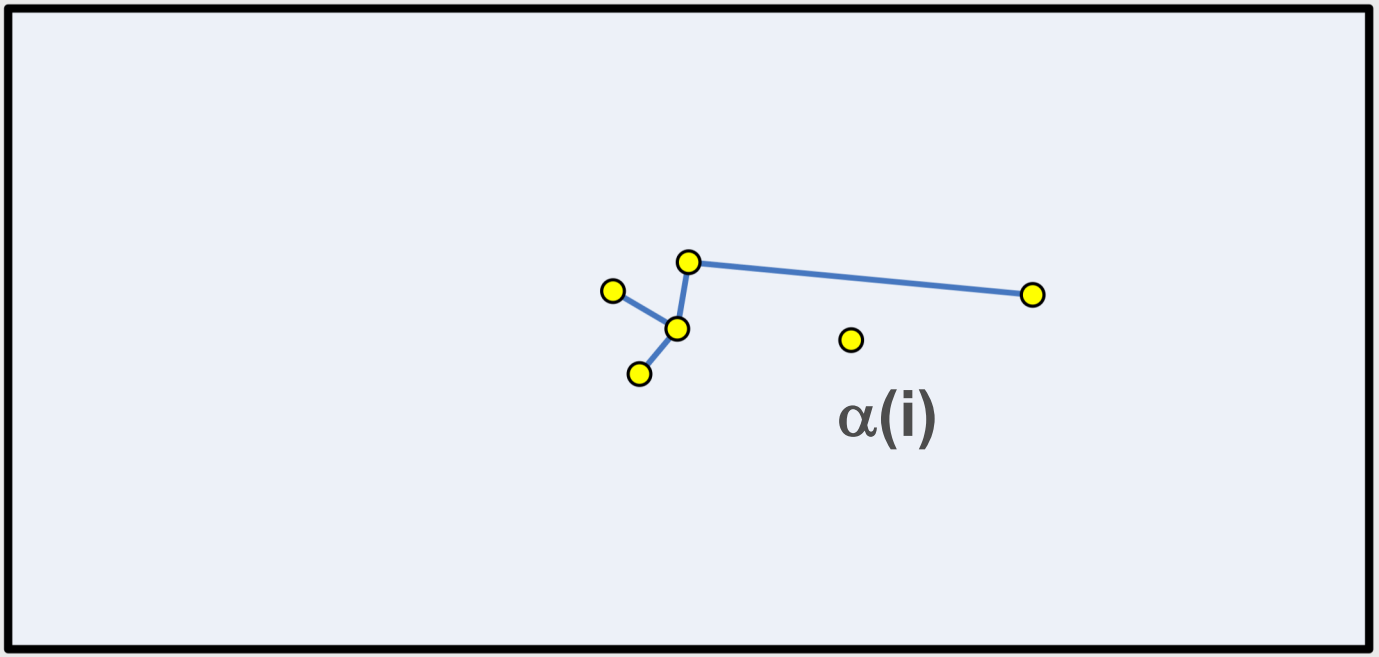
Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));



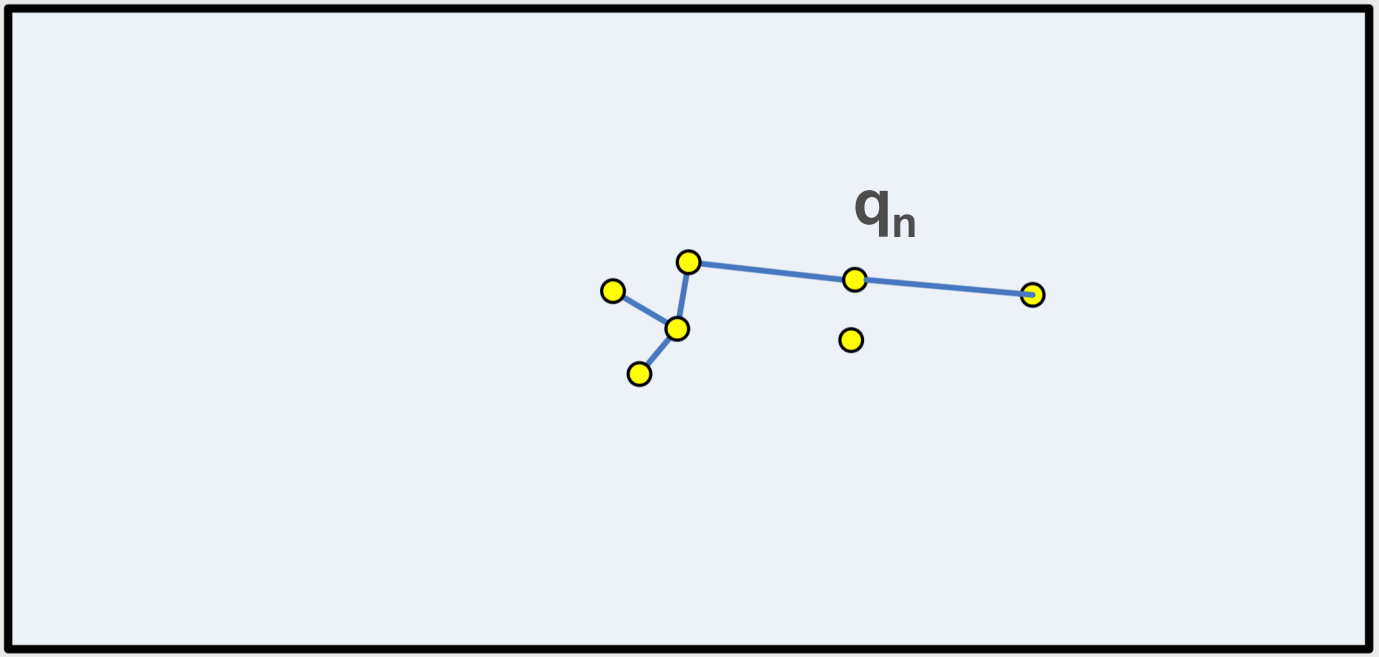
Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));



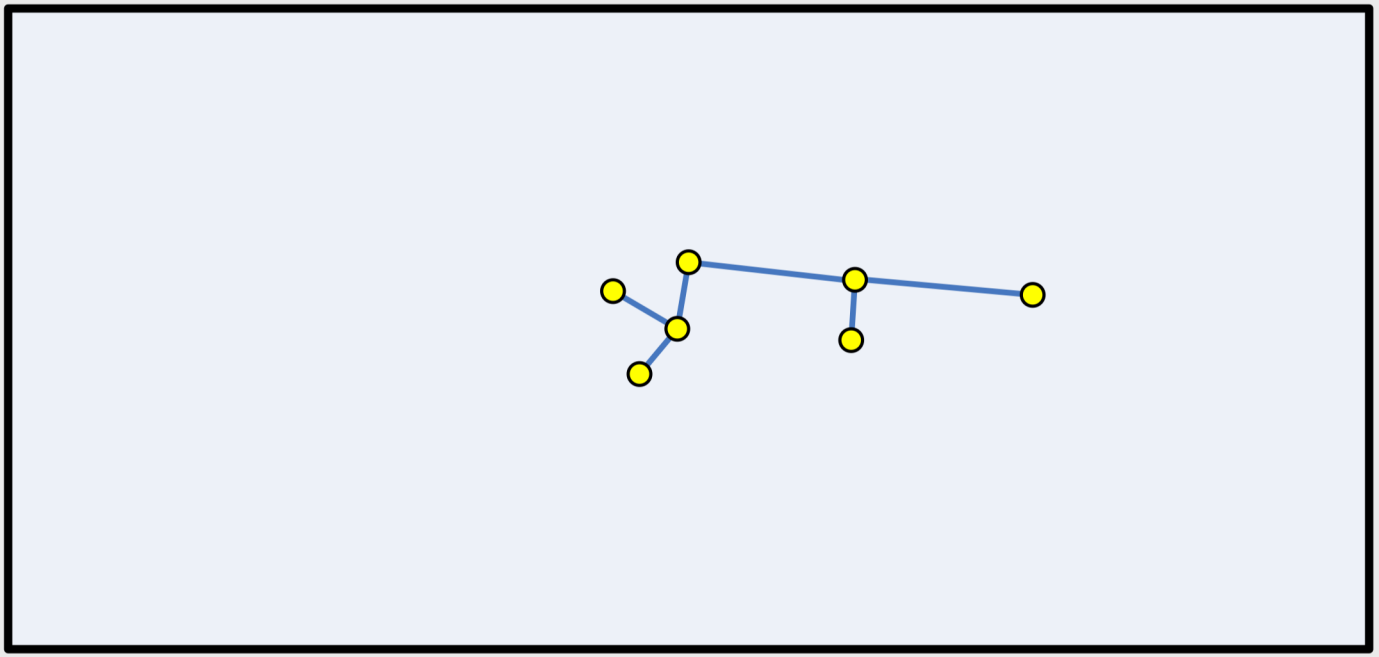
Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(t restricted to points)
- 5 G.add_edge(q_n; α(i));



Example: Simplest one

- SIMPLE RDT(q_0)
- 1 G.init(q_0);
- 2 for i = 1 to k do
- 3 G.add_vertex(α(i));
- 4 q_n ← nearest(S(G); α(i));
- 5 G.add_edge(q_n; α(i));

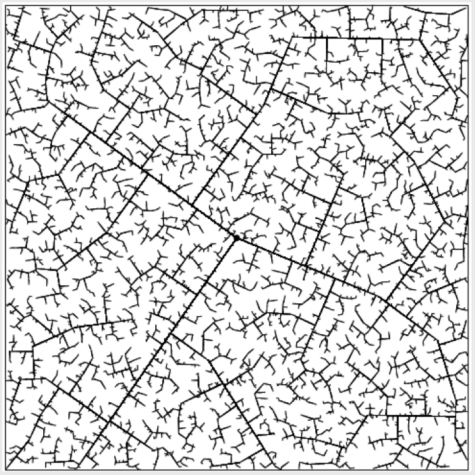


Example: Simplest one / Summary

- Every iteration the tree grows iteratively by connecting $\alpha(i)$ to S.
- In every iteration $\alpha(i)$ becomes a vertex → tree is dense

$C = [0; 1]^2$,
$q_0 = (1/2; 1/2)$.

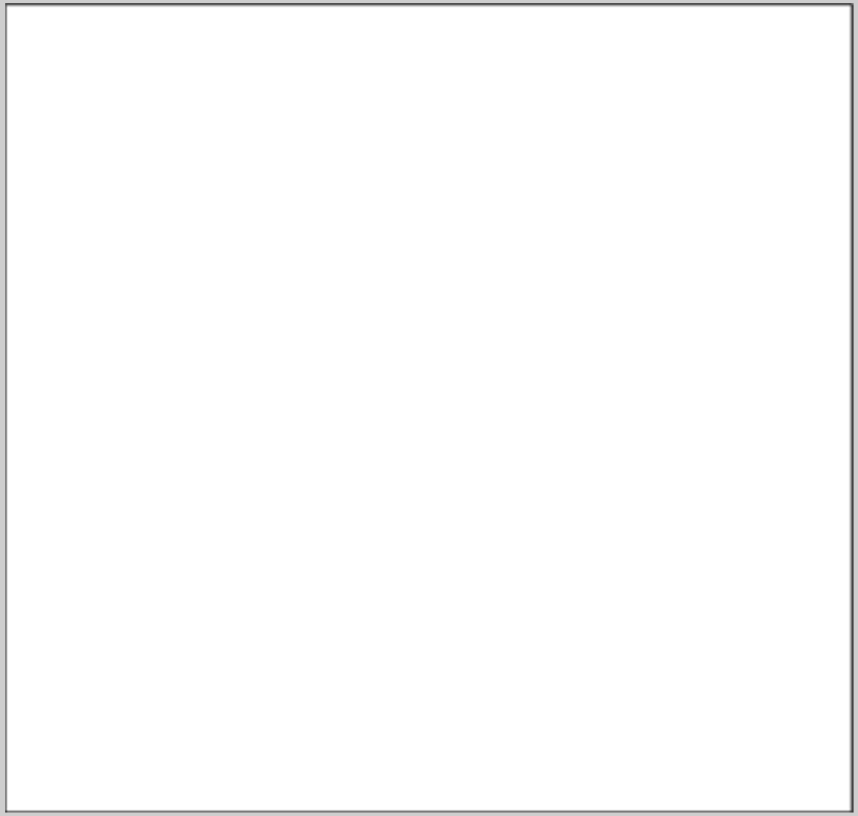
45 iterations



2345 iterations

- RRT quickly reaches the unexplored parts.
- RRT is dense → it gets arbitrarily to any point in the space.

Example: RDT

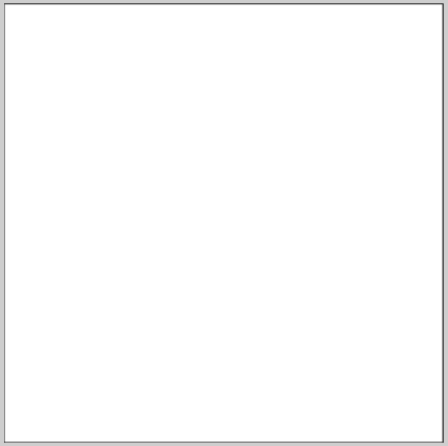
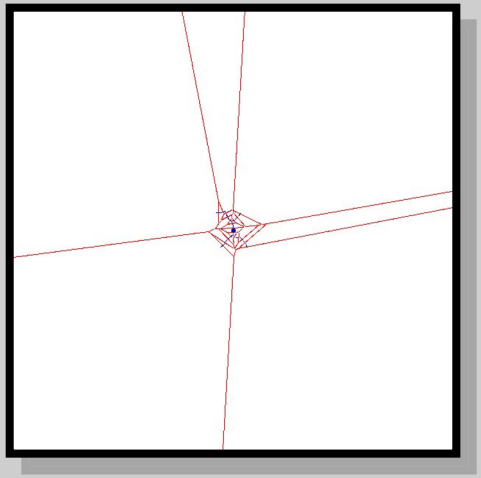
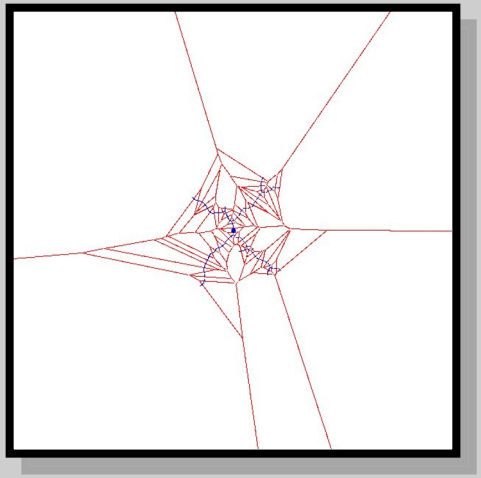
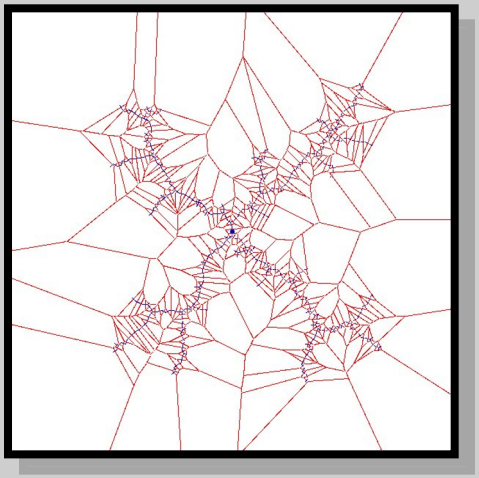
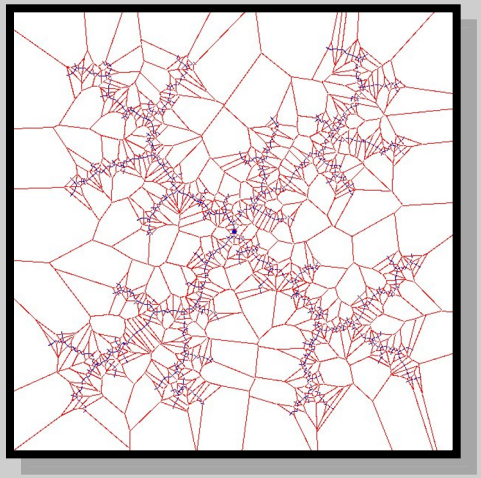
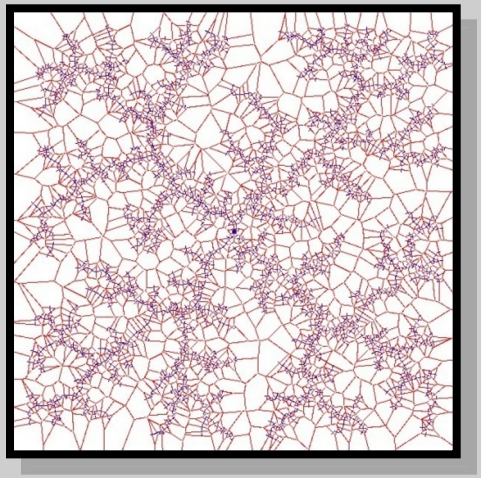


RDT: properties / features

- First C-Space is examined into every direction → rapidly reaches corners
- Gradually, C-Space is filled up with finer branches
- In the end, C-Space is completely covered

- Because of tree gradually increasing resolution → perfect suited for sampling based motion planning

RDT and Voronoi

Using RDT for path planning

- Simple tree search
- Use proposed algorithm that takes ostacles into account
- Periodically add goal (e.g. every 100th step) and test whether it is reachable