15-780: Grad AI
Lecture 16: Probability

Geoff Gordon (this lecture)
Tuomas Sandholm
TAs Erik Zawadzki, Abe Othman

Randomness in search

Rapidly-exploring Random Trees

- Break up C-space into Voronoi regions around random landmarks
- Invariant: landmarks always form a tree
  - known path to root
- Subject to this requirement, placed in a way that tends to split large Voronoi regions
  - coarse-to-fine search
- Goal: ***feasibility*** not ***optimality*** *(*)*

RRT: required subroutines

- RANDOM_CONFIG
  - samples from C-space
- EXTEND(**q**, **q**')
  - local controller, heads toward **q**' from **q**
  - stops before hitting obstacle (and perhaps also after bound on time or distance)
- FIND_NEAREST(**q**, Q)
  - searches current tree Q for point near **q**

Path Planning with RRTs

RRT = Rapidly-Exploring Random Tree

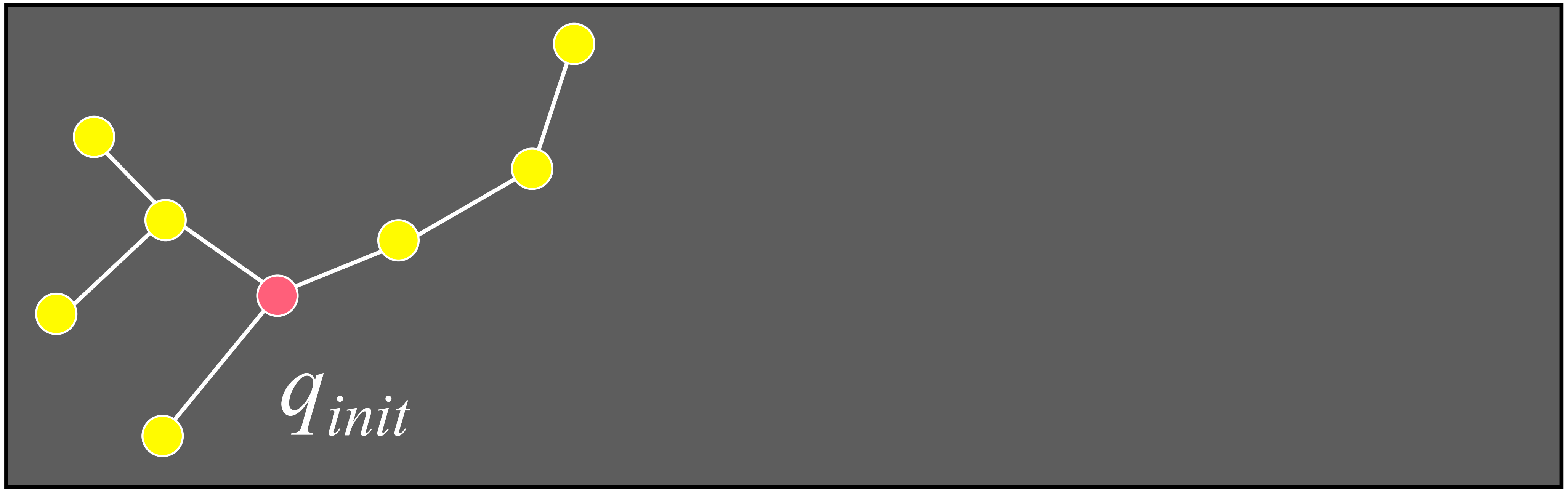
```
BUILT_RRT($q_{init}$) {
    T = $q_{init}$
    for k = 1 to K {
        $q_{rand}$ = RANDOM_CONFIG()
        EXTEND(T, $q_{rand}$);
    }
}
```

```
EXTEND(T, q) {
    $q_{near}$ = FIND_NEAREST(q, T)
    $q_{new}$ = EXTEND($q_{near}$, q)
    T = T + ($q_{near}$, $q_{new}$)
}
```

[Kuffner & LaValle , ICRA'00]

Path Planning with RRTs

RRT = Rapidly-Exploring Random Tree



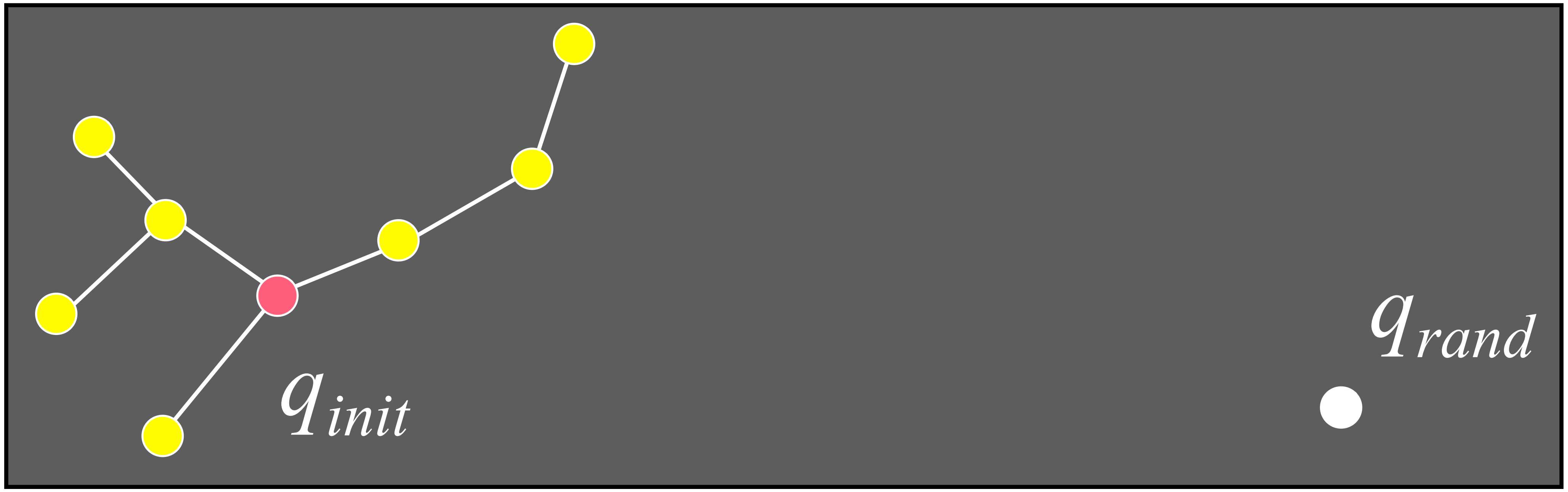
```
BUILT_RRT($q_{init}$) {
    T = $q_{init}$
    for k = 1 to K {
        $q_{rand}$ = RANDOM_CONFIG()
        EXTEND(T, $q_{rand}$);
    }
}
```

```
EXTEND(T, q) {
    $q_{near}$ = FIND_NEAREST(q, T)
    $q_{new}$ = EXTEND($q_{near}$, q)
    T = T + ($q_{near}$, $q_{new}$)
}
```

[Kuffner & LaValle , ICRA’00]

Path Planning with RRTs

RRT = Rapidly-Exploring Random Tree



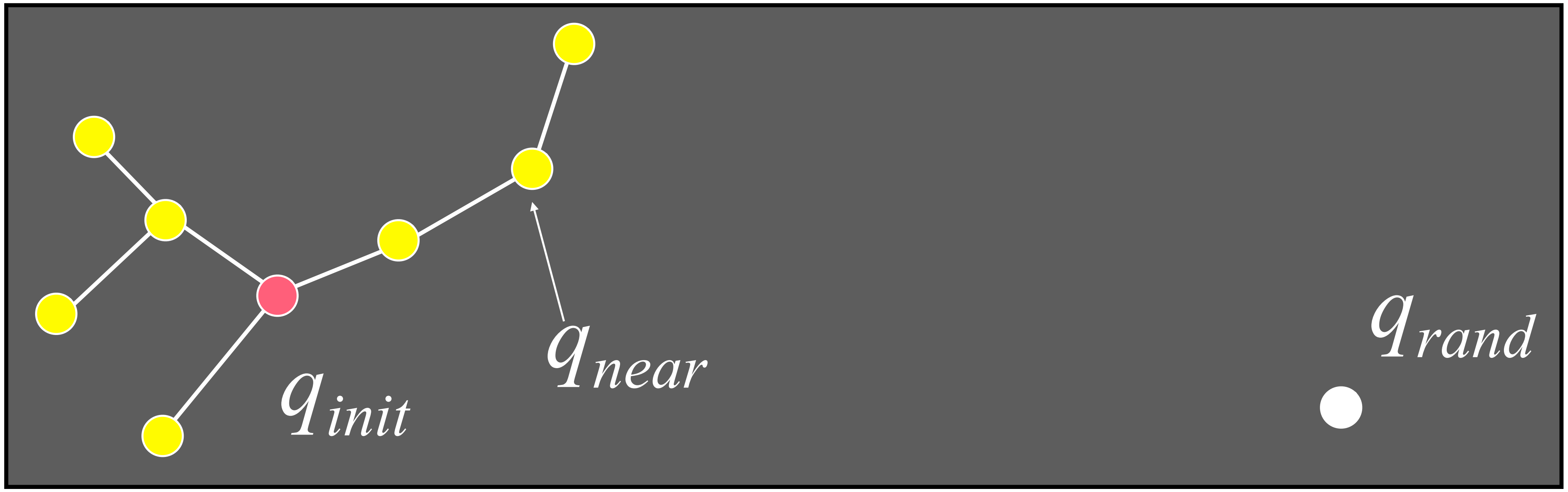
```
BUILT_RRT($q_{init}$) {
    T = $q_{init}$
    for k = 1 to K {
        $q_{rand}$ = RANDOM_CONFIG()
        EXTEND(T, $q_{rand}$);
    }
}
```

```
EXTEND(T, q) {
    $q_{near}$ = FIND_NEAREST(q, T)
    $q_{new}$ = EXTEND($q_{near}$, q)
    T = T + ($q_{near}$, $q_{new}$)
}
```

[Kuffner & LaValle , ICRA’00]

Path Planning with RRTs

RRT = Rapidly-Exploring Random Tree



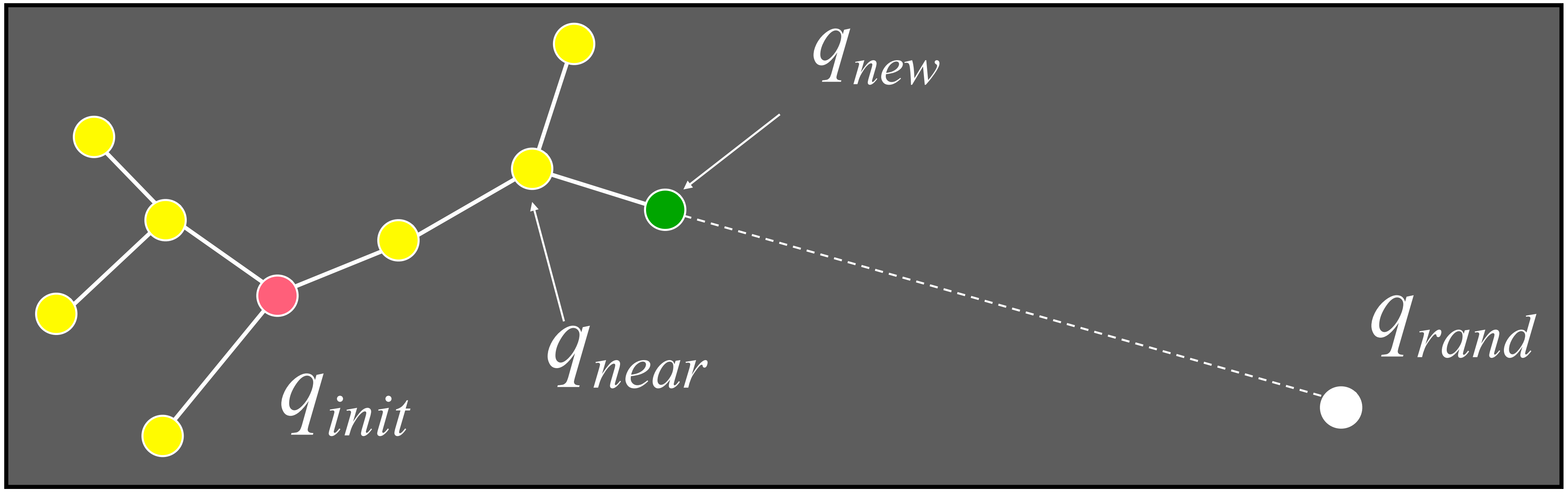
```
BUILT_RRT($q_{init}$) {
    T = $q_{init}$
    for k = 1 to K {
        $q_{rand}$ = RANDOM_CONFIG()
        EXTEND(T, $q_{rand}$);
    }
}
```

```
EXTEND(T, q) {
    $q_{near}$ = FIND_NEAREST(q, T)
    $q_{new}$ = EXTEND($q_{near}$, q)
    T = T + ($q_{near}$, $q_{new}$)
}
```

[Kuffner & LaValle , ICRA’00]

Path Planning with RRTs

RRT = Rapidly-Exploring Random Tree

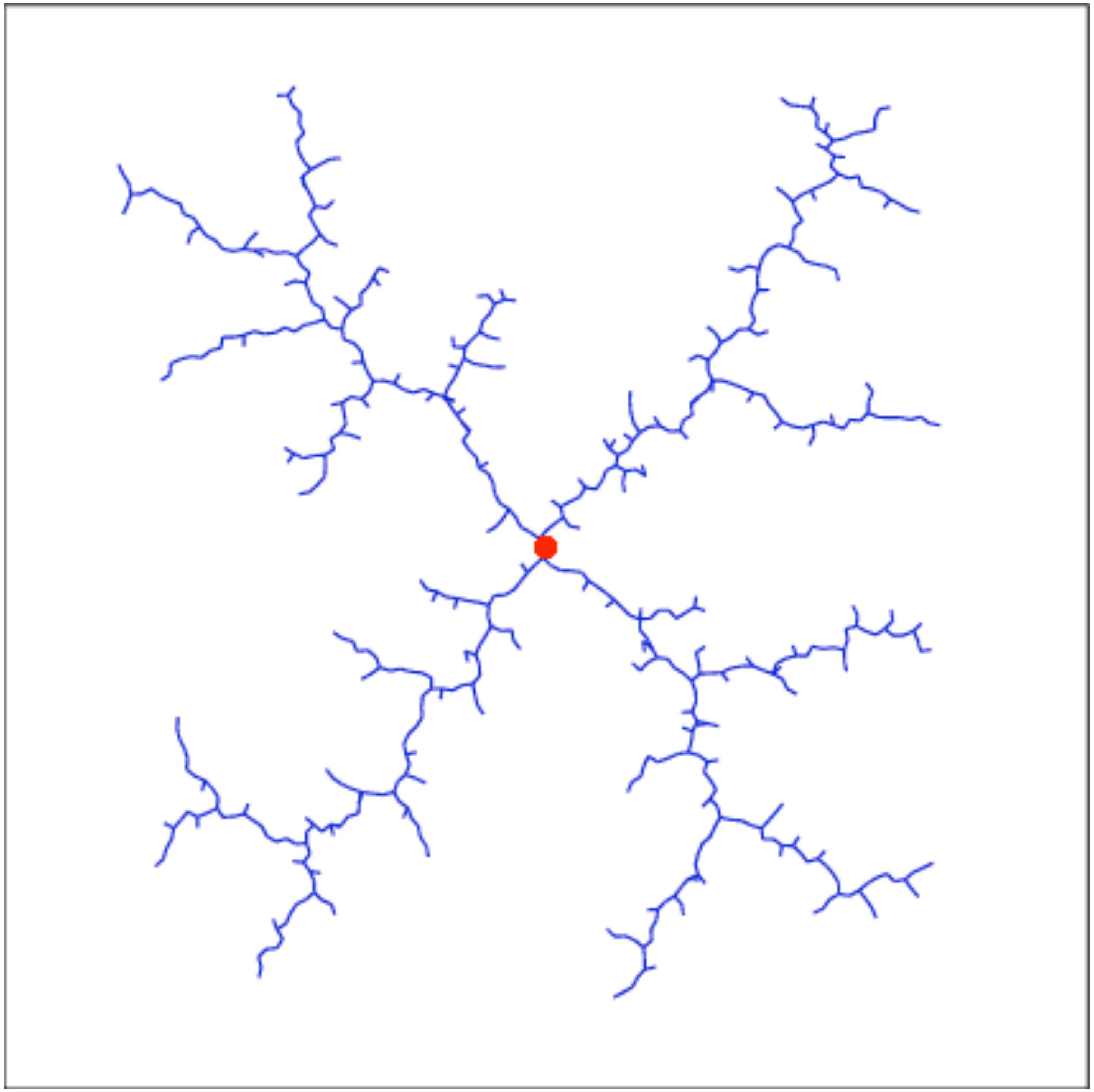


```
BUILT_RRT($q_{init}$) {
    T = $q_{init}$
    for k = 1 to K {
        $q_{rand}$ = RANDOM_CONFIG()
        EXTEND(T, $q_{rand}$);
    }
}
```

```
EXTEND(T, q) {
    $q_{near}$ = FIND_NEAREST(q, T)
    $q_{new}$ = EXTEND($q_{near}$, q)
    T = T + ($q_{near}$, $q_{new}$)
}
```

[Kuffner & LaValle , ICRA'00]

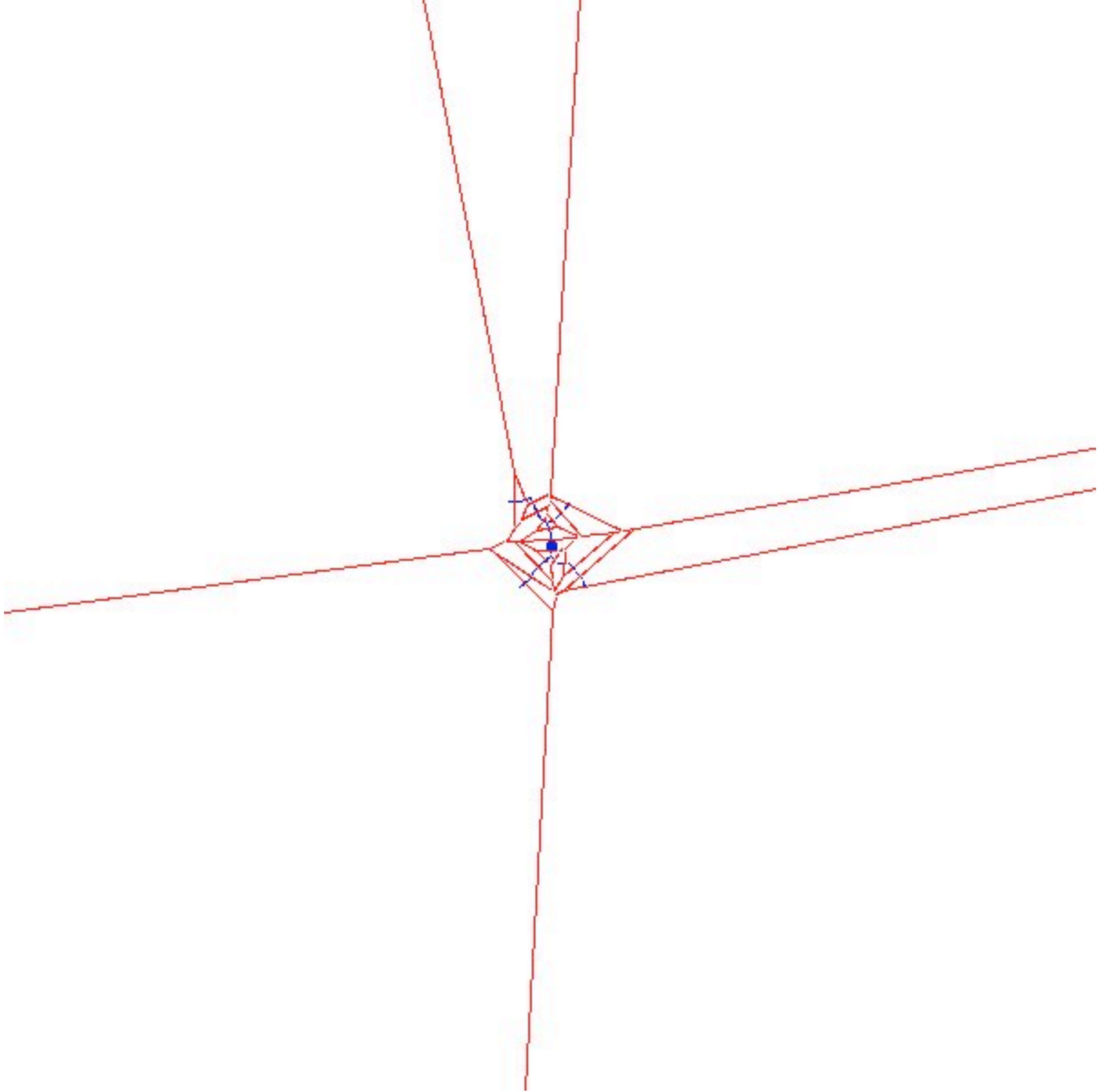
RRT example

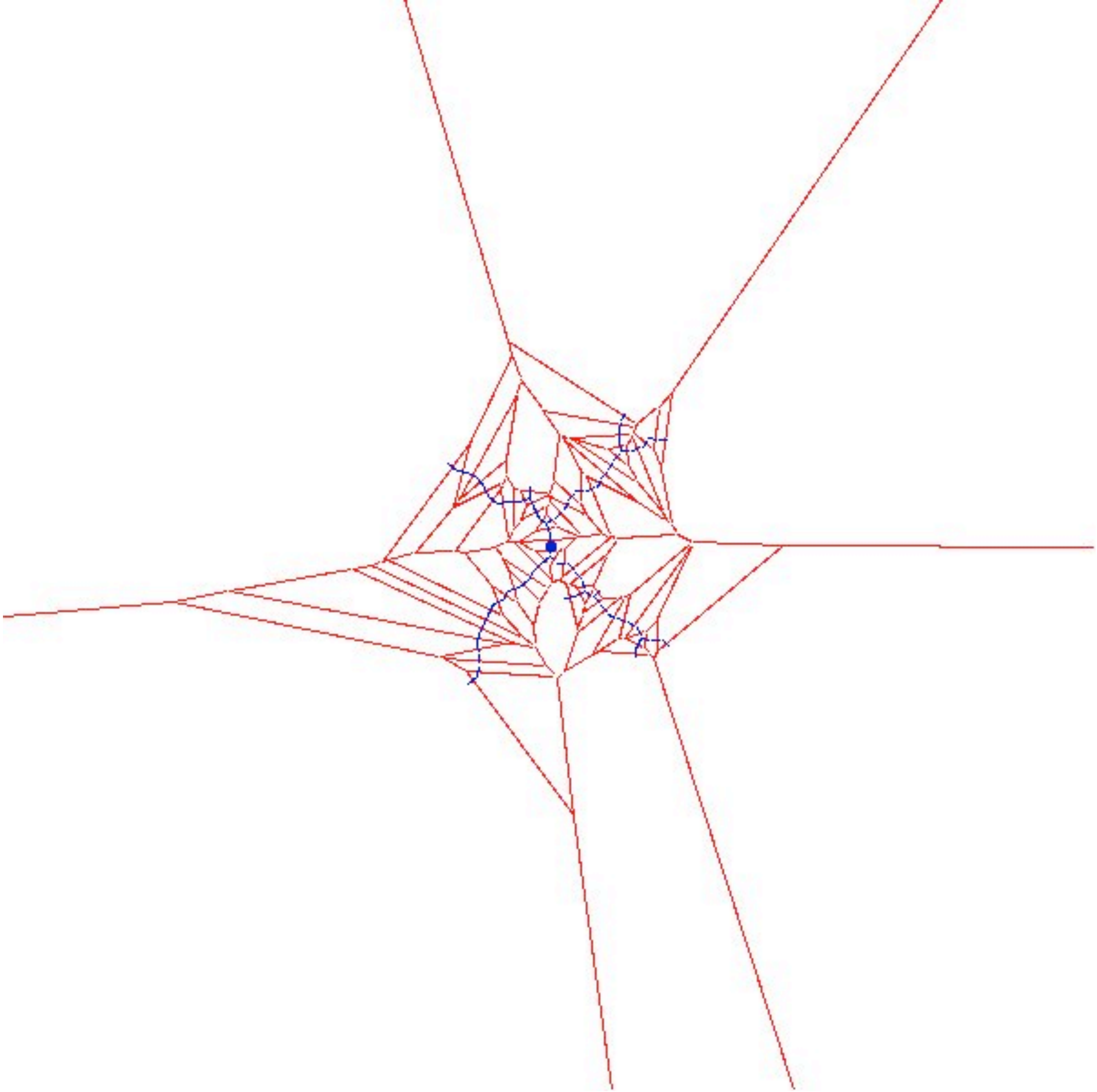


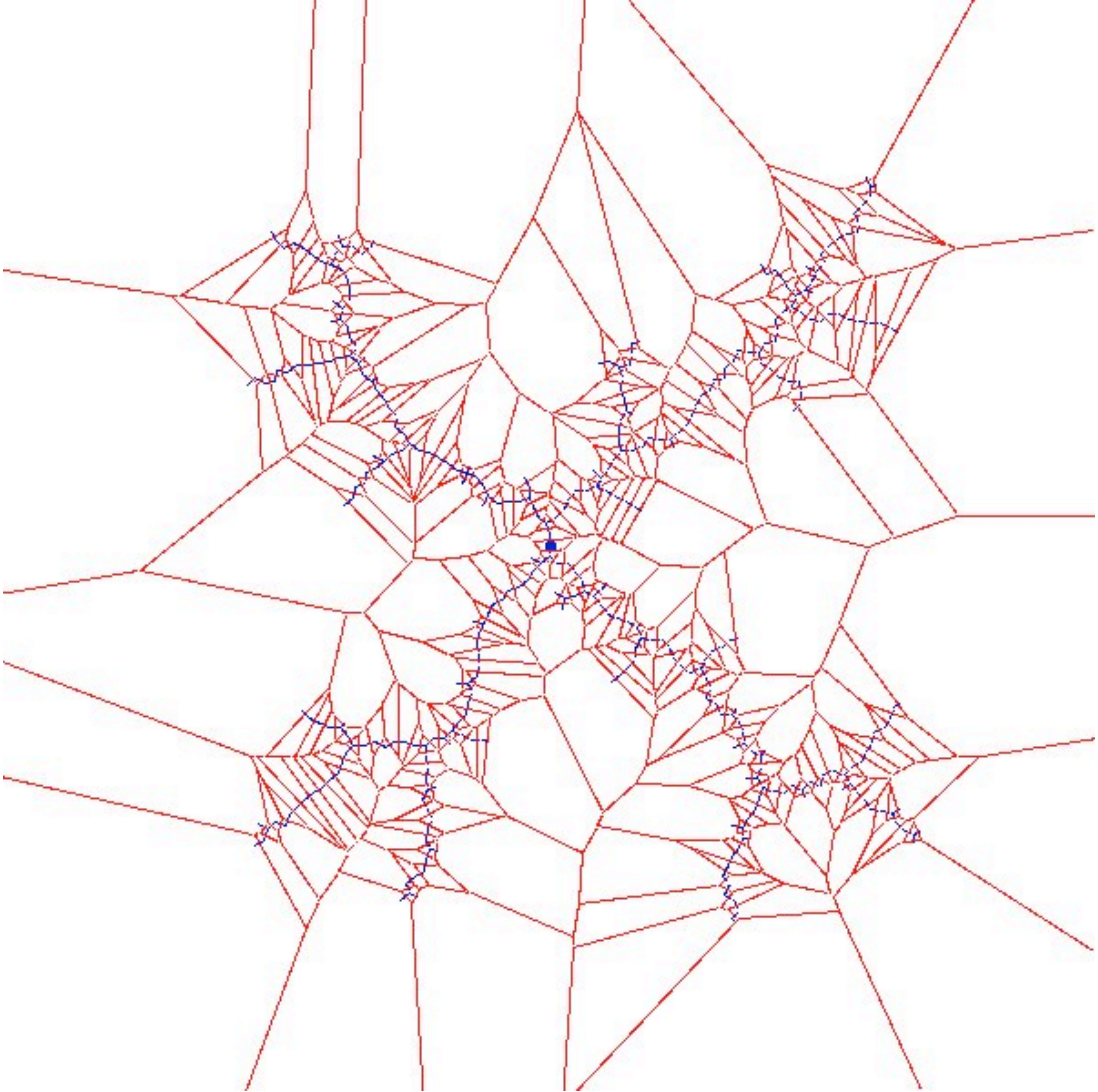
Planar holonomic robot

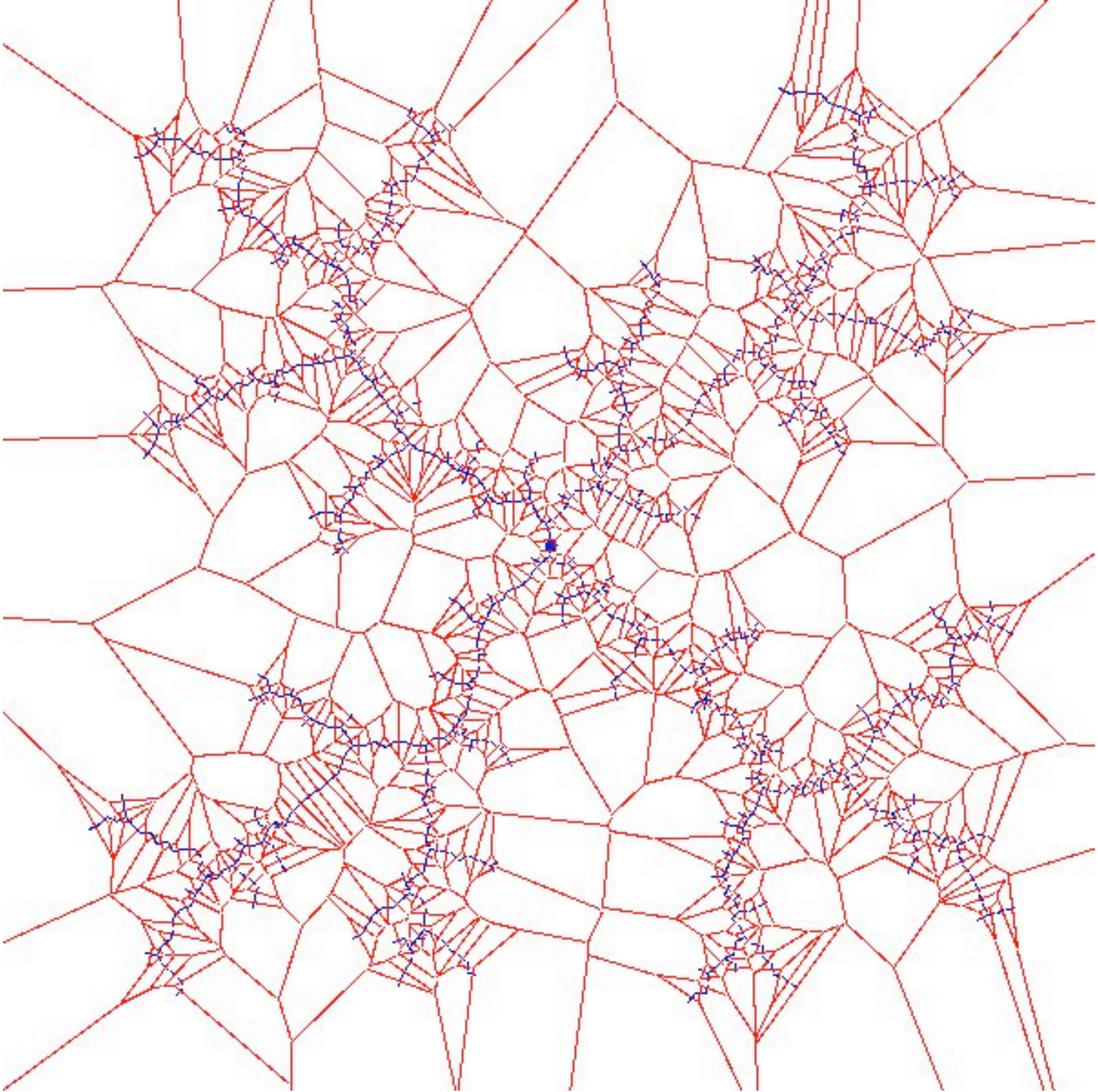
RRTs explore coarse to fine

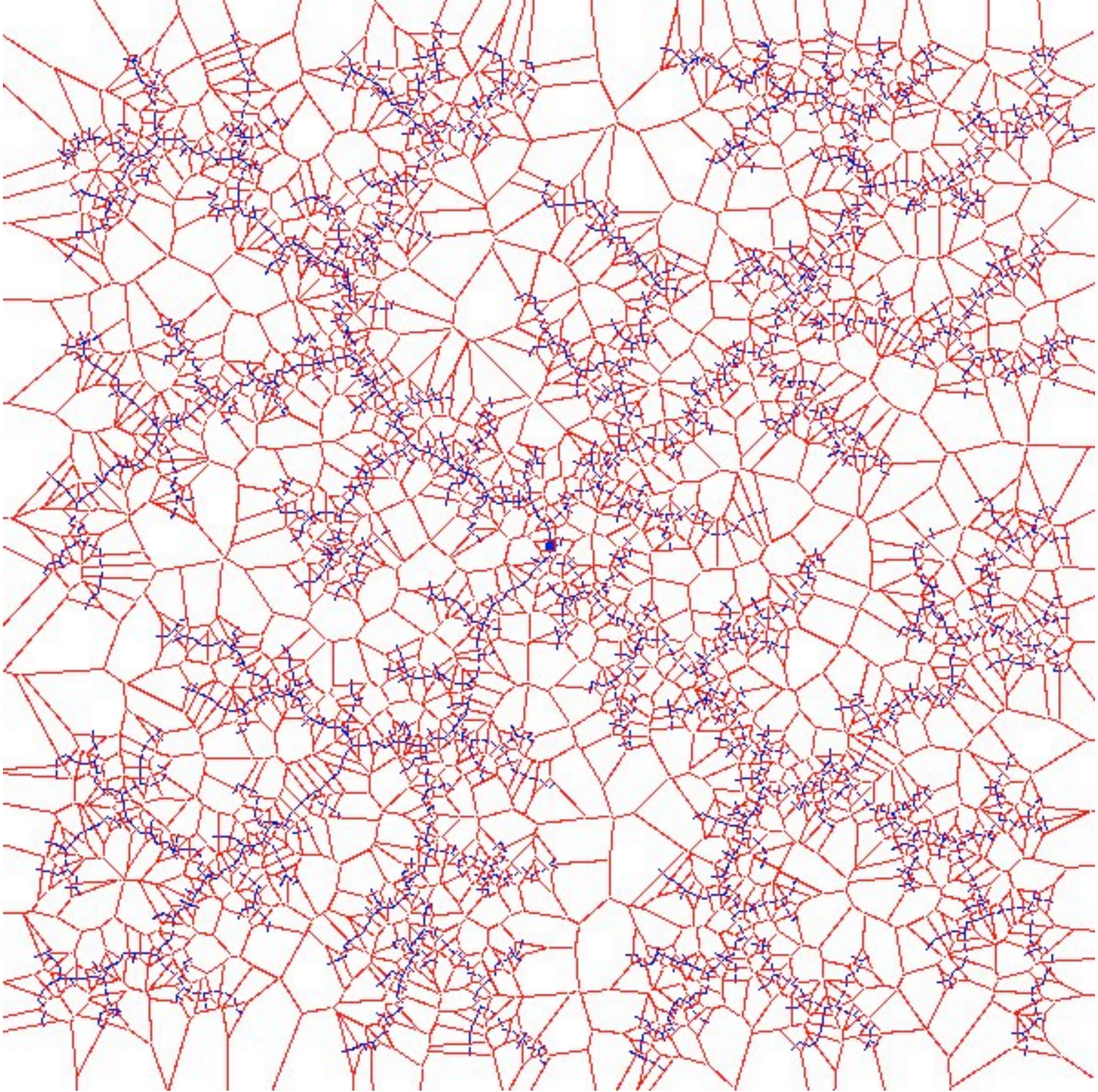
- Tend to break up large Voronoi regions
  - higher probability of q_{rand} being in them
- Limiting distribution of vertices given by RANDOM_CONFIG
  - as RRT grows, probability that q_{rand} is reachable with local controller (and so immediately becomes a new vertex) approaches 1



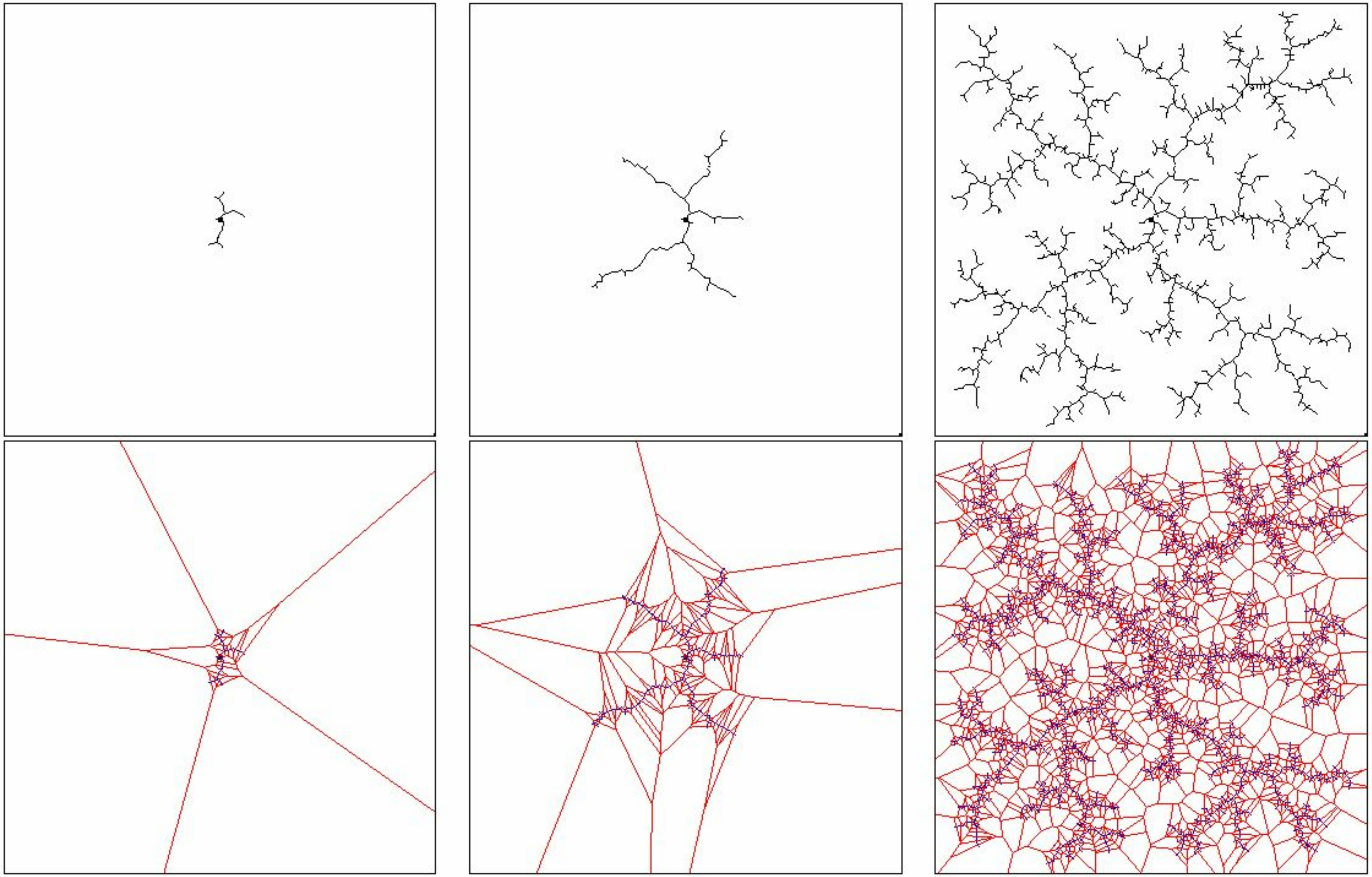




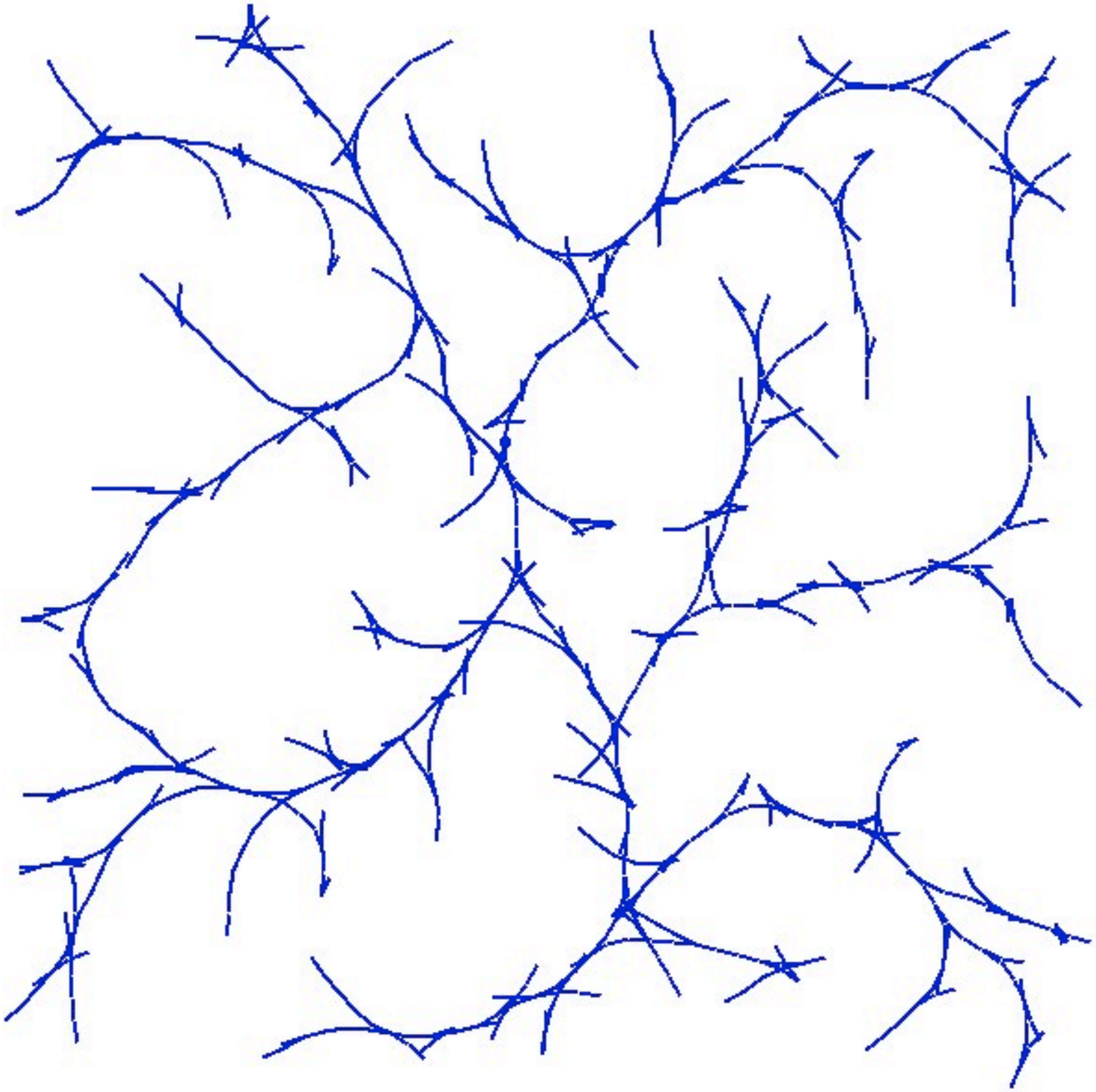




RRT example

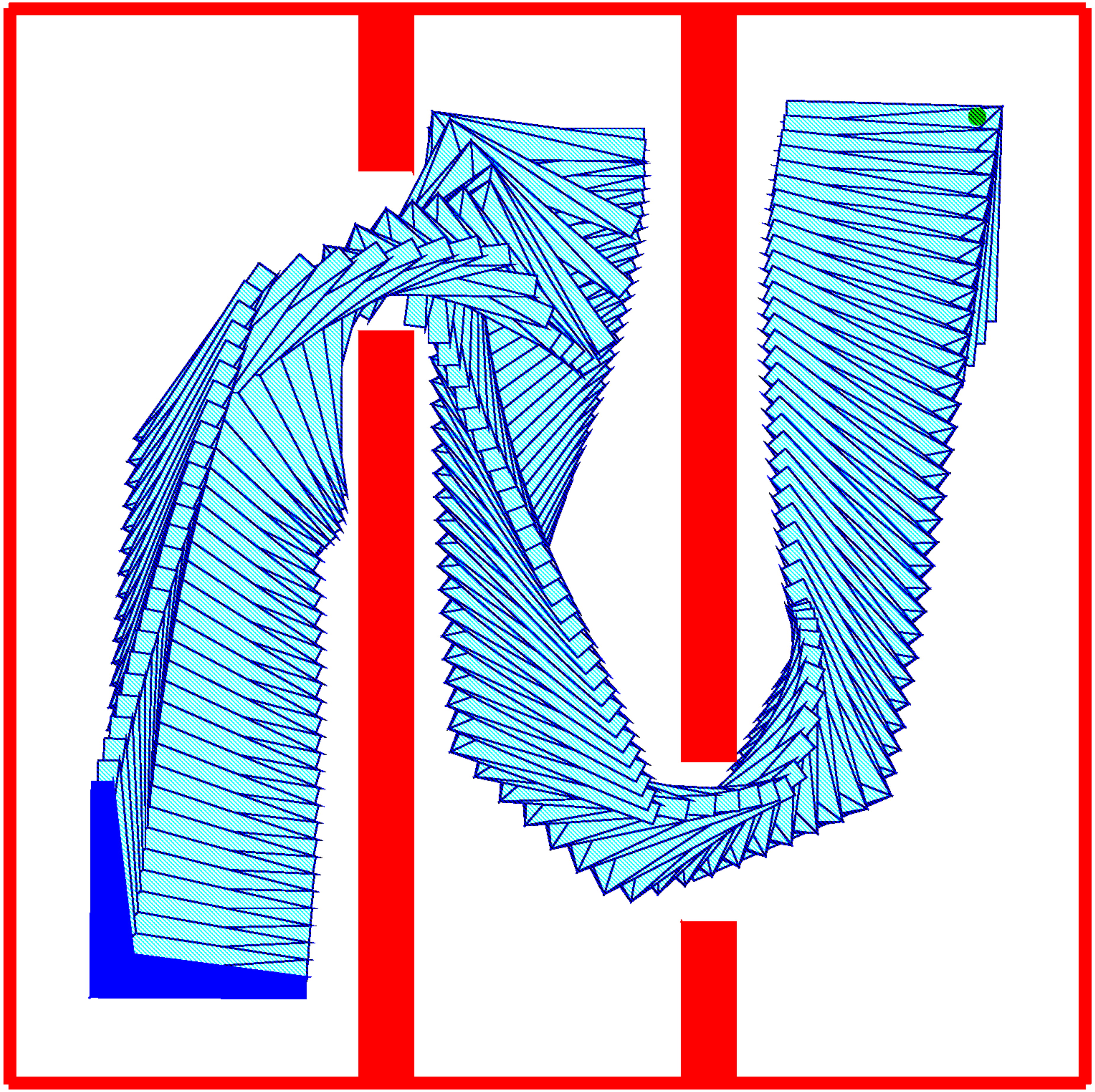


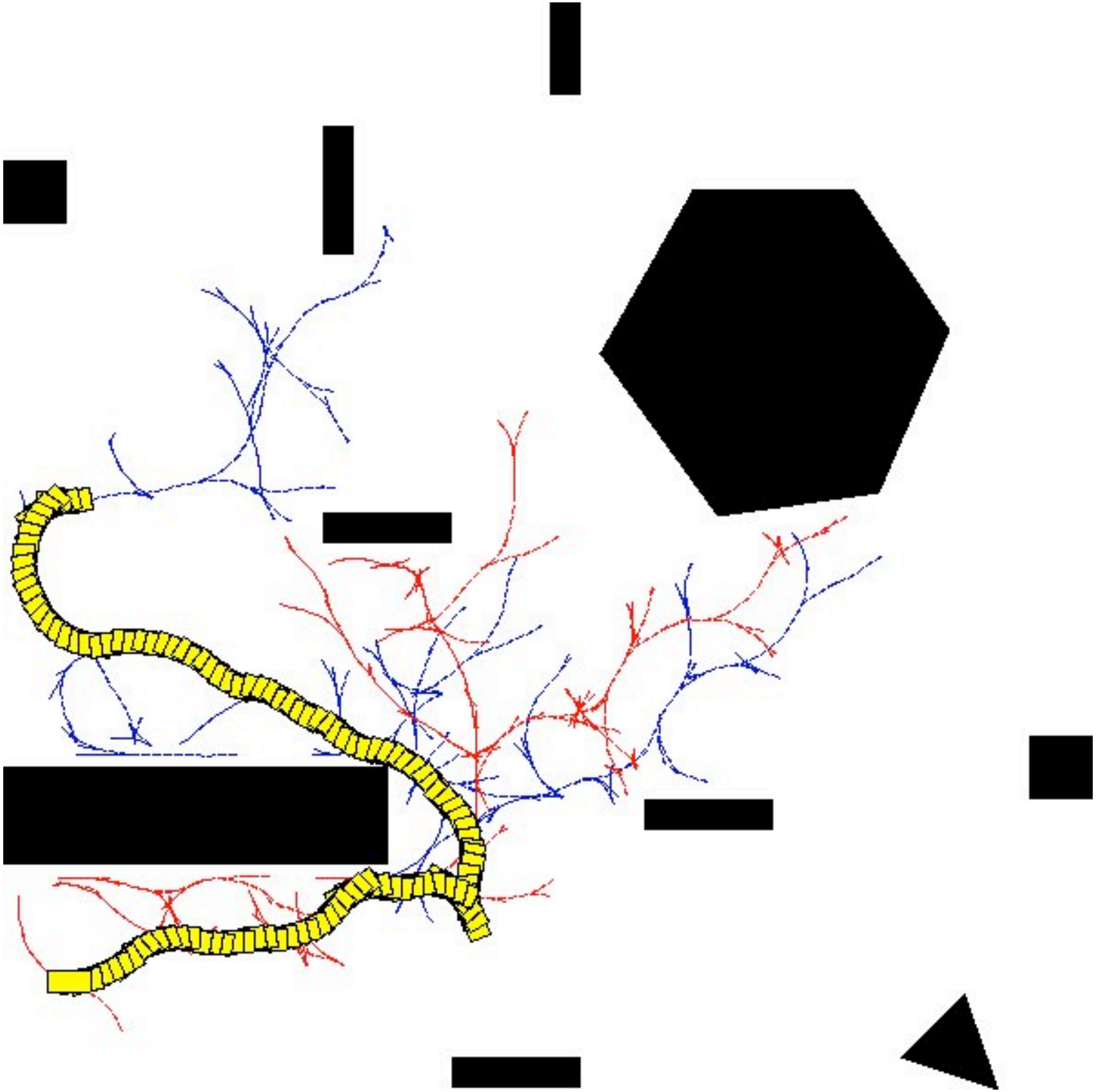
RRT for a car (3 dof)

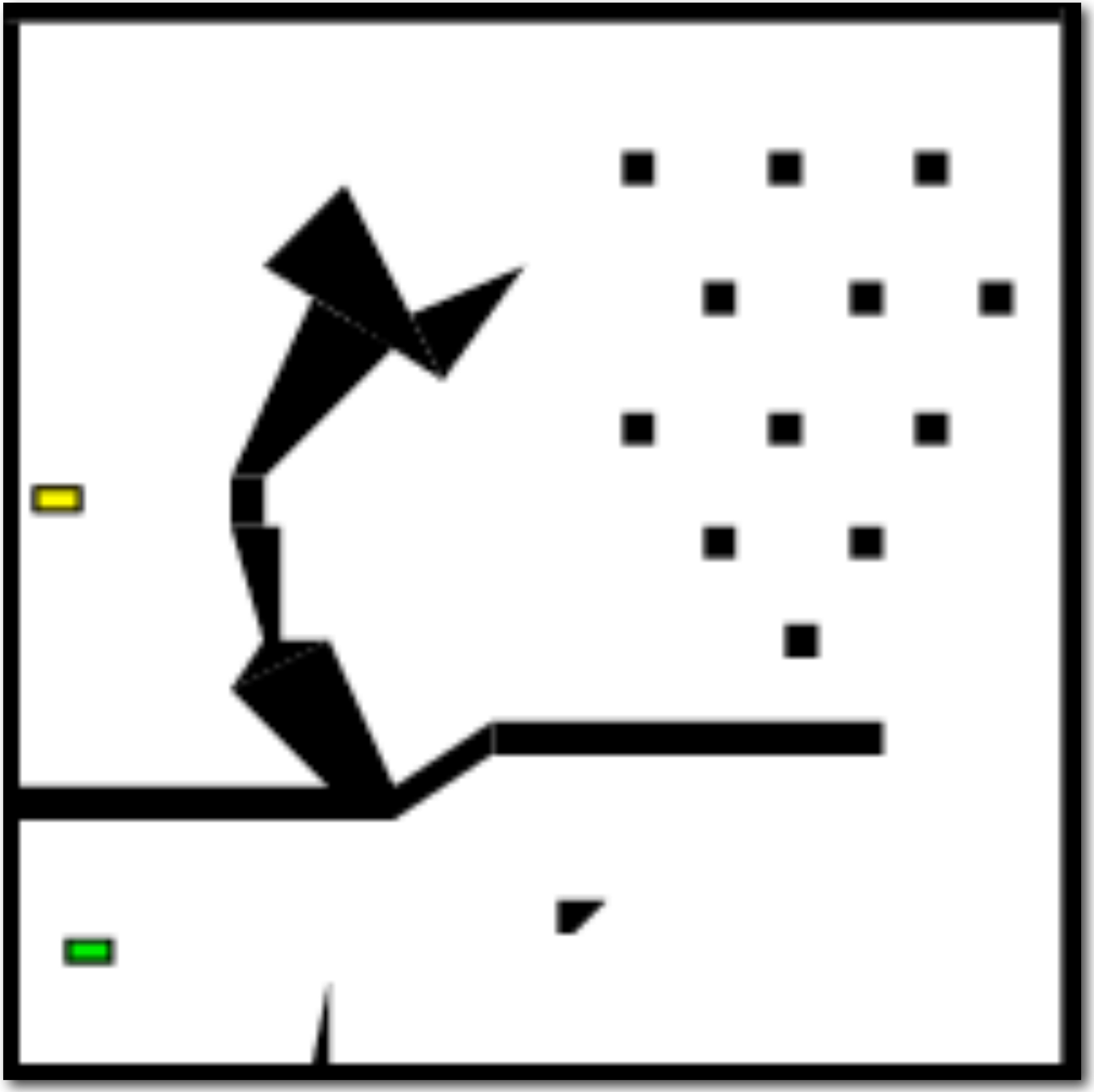


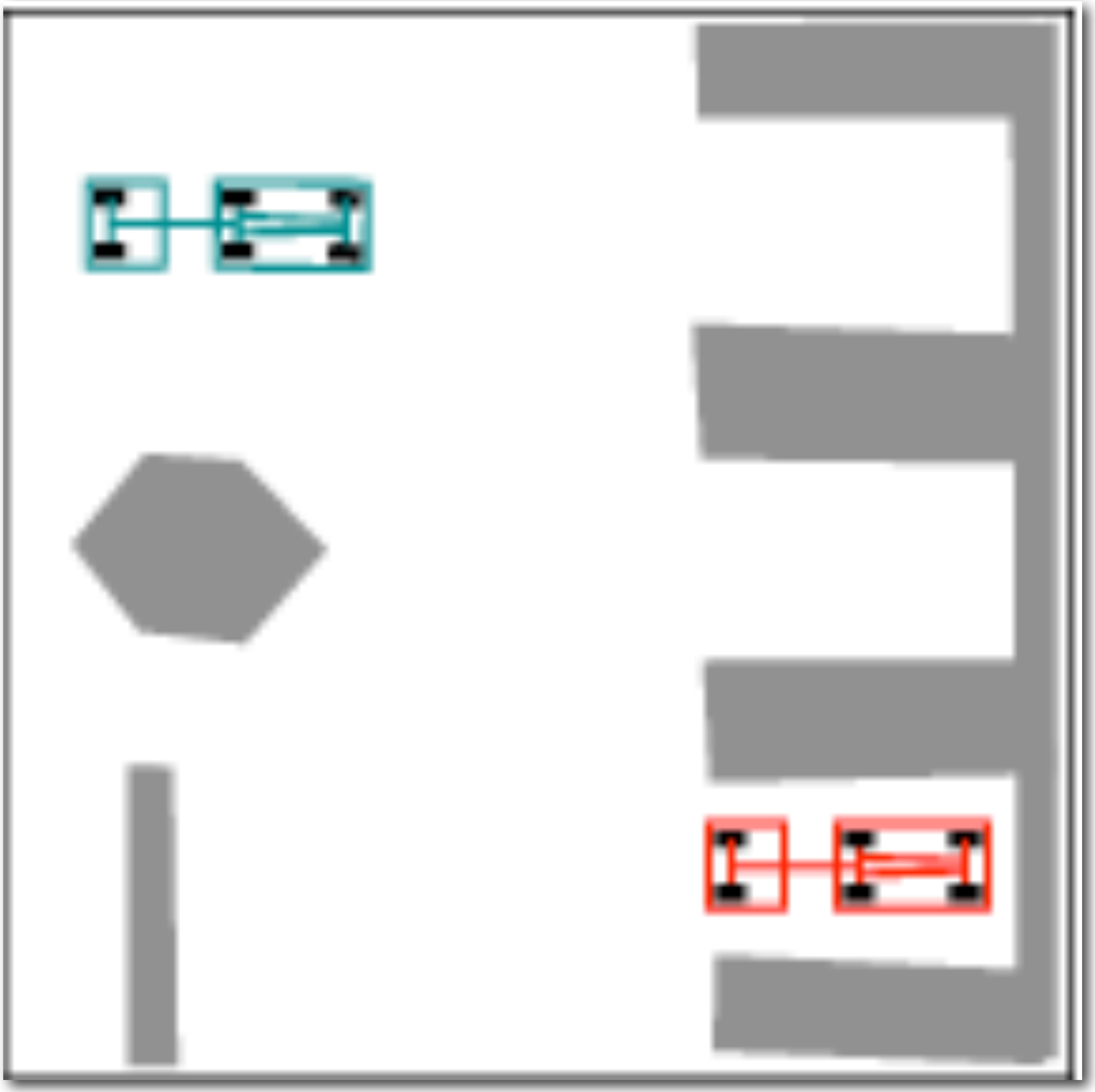
Planning with RRTs

- Build RRT from start until we add a node that can reach goal using local controller
- (Unique) path: root → last node → goal
- Optional: “rewire” tree during growth by testing connectivity to more than just closest node
- Optional: grow forward and backward









Probability

Probability

- Random variables

- Atomic events

- Sample space

Probability

- Events

- Combining events

Probability

- Measure:
  - disjoint union:
  - e.g.:
  - interpretation:
- Distribution:
  - interpretation:
  - e.g.:

Example

		AAPL price		
		up	same	down
Weather	sun	0.09	0.15	0.06
	rain	0.21	0.35	0.14

Bigger example

AAPL price

			up	same	dow
PIT	Weather	sun	0.03	0.05	0.02
		rain	0.07	0.12	0.05

			up	same	dow
LAX	Weather	sun	0.14	0.23	0.09
		rain	0.06	0.10	0.04

Notation

- $X=x$: event that r.v. X is realized as value x
- $P(X=x)$ means probability of event $X=x$
  - if clear from context, may omit "$X=$"
  - instead of $P(\text{Weather}=\text{rain})$, just $P(\text{rain})$
  - complex events too: e.g., $P(X=x, Y\neq y)$
- $P(X)$ means a function: $x \rightarrow P(X=x)$

Functions of RVs

- Extend definition: any deterministic function of RVs is also an RV
- E.g., “profit”:

		AAPL price		
		up	same	down
Weather	sun	–11	0	11
	rain	–11	0	11

Sample v. population

- Suppose we watch for 100 days and count up our observations

		AAPL price		
		up	same	dow
Weather	sun	0.09	0.15	0.06
	rain	0.21	0.35	0.14

		AAPL price		
		up	same	dow
Weather	sun	7	12	3
	rain	22	41	15

Law of large numbers

(simple version)

- If we take a sample of size N from distribution P, count up frequencies of atomic events, and normalize (divide by N) to get a distribution $\tilde{P}$
- Then $\tilde{P} \rightarrow P$ as $N \rightarrow \infty$

Working w/ distributions

- Marginals (eliminate an irrelevant RV)
- Conditionals (incorporate an observation)
- Joint (before marginalizing or conditioning)

Marginals

		AAPL price		
		up	same	down
Weather	sun	0.09	0.15	0.06
	rain	0.21	0.35	0.14

Law of total probability

also called "sum rule"

- Two RVs, X and Y
- Y has values $y_1, y_2, \ldots, y_k$
- $P(X) = P(X, Y=y_1) + P(X, Y=y_2) + \ldots$

Conditioning on an observation

	Coin	
Weather	H	T
sun	0.15	0.15
rain	0.35	0.35

- Two steps:
  - enforce consistency
  - renormalize
- Notation:

Conditionals

AAPL price

PIT Weather

	up	same	down
sun	0.03	0.05	0.02
rain	0.07	0.12	0.05

LAX Weather

	up	same	down
sun	0.14	0.23	0.09
rain	0.06	0.10	0.04

Conditionals

- Thought experiment: what happens if we condition on an event of zero probability?

Notation

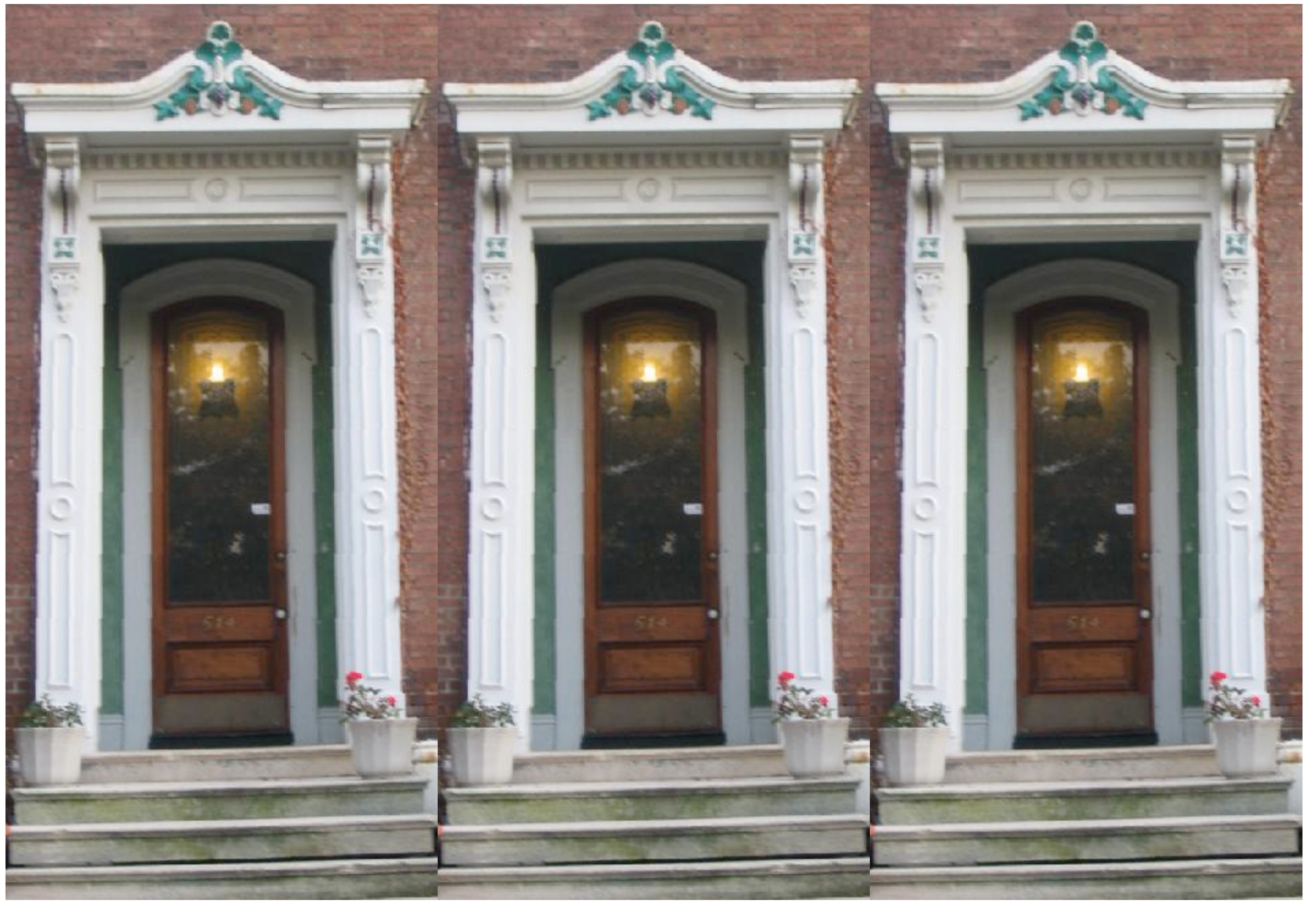
- $P(X \mid Y)$ is a function: x, y → $P(X=x \mid Y=y)$
- So:
  - $P(X \mid Y)\ P(Y)$ means the function
    x, y →

Conditionals in literature

When you have eliminated the impossible, whatever remains, however improbable, must be the truth.

—*Sir Arthur Conan Doyle, as Sherlock Holmes*

Exercise



$$$

Exercise