Real-Time Motion Planning and Autonomous Driving

Jeffrey Ichnowski

What is Real-Time Motion Planning?

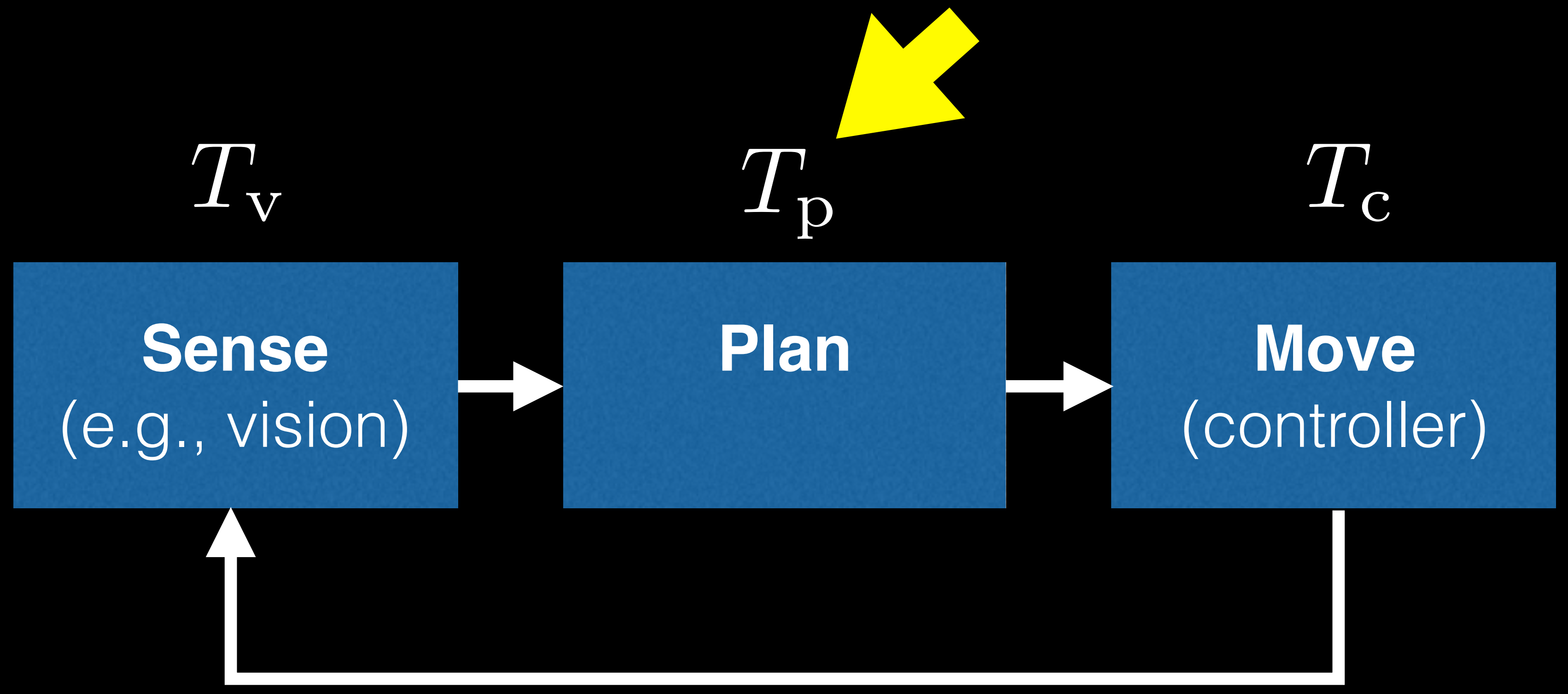
1st floor vs. 2nd floor

Motion planning with a hard real-time constraint.

temporally correct

$$T_i = (\phi_i, p_i, e_i, D_i)$$

Example Real-Time Motion Planning System

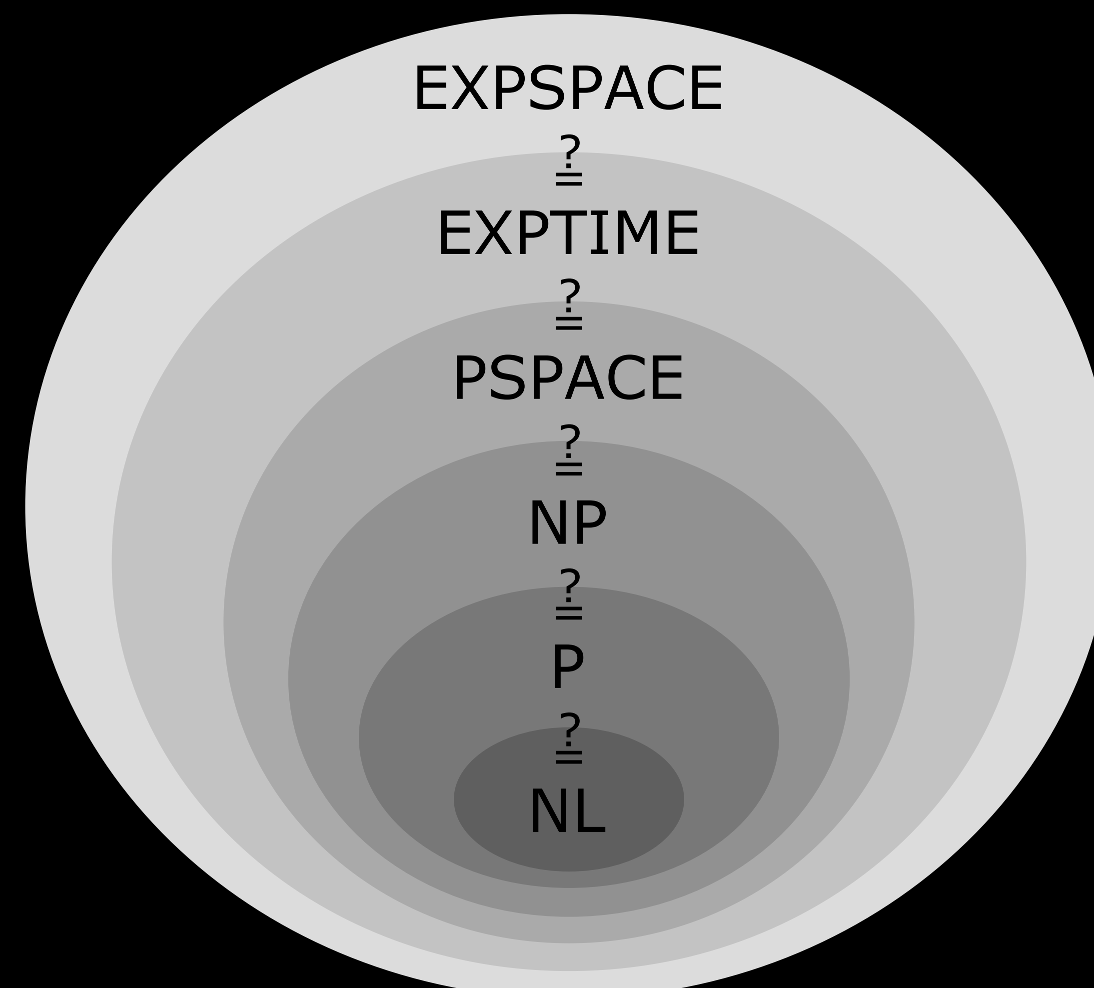


Problems with Motion Planning in a Real-Time System

motion planning P-SPACE hard

exponential in dimensions

uncountably infinite
state spaces



[Image source: wikipedia]

Real-Time Motion Planning

In the general case: impossible.

Precompute Motion Plan

Extremely popular option.

- Allows arbitrarily long computation
- Asymptotically feasible algorithms
- Asymptotically optimal algorithms

Is this real-time? yes and no.

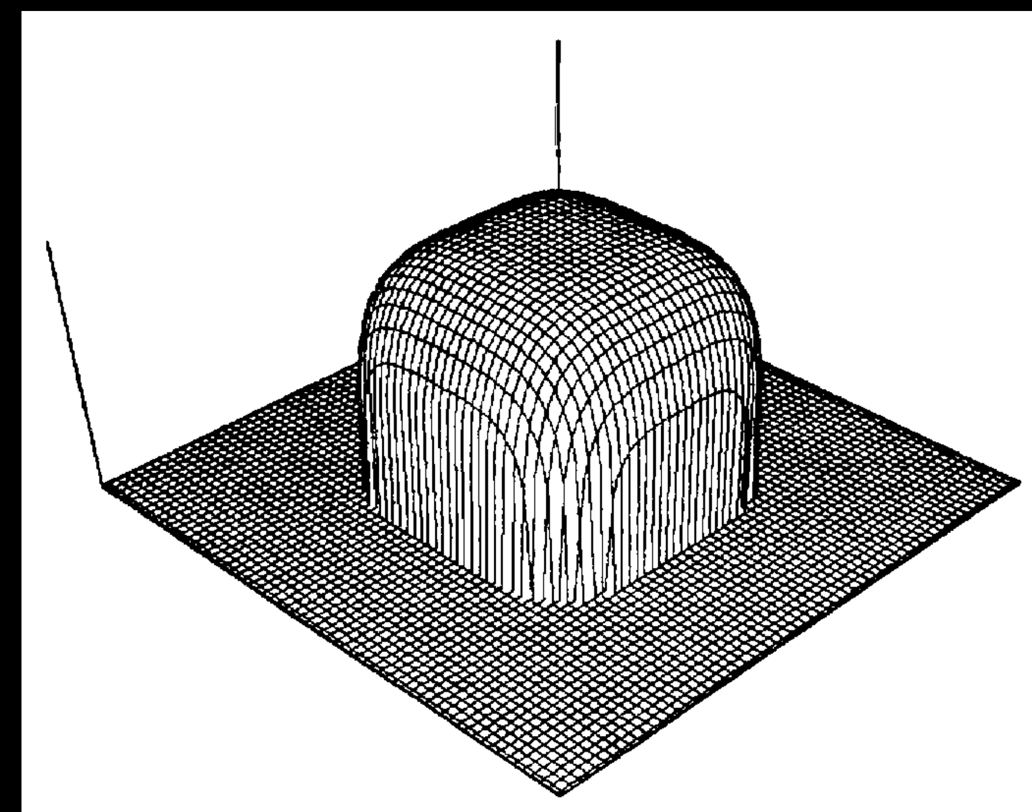
Real-time Motion Planning Problem

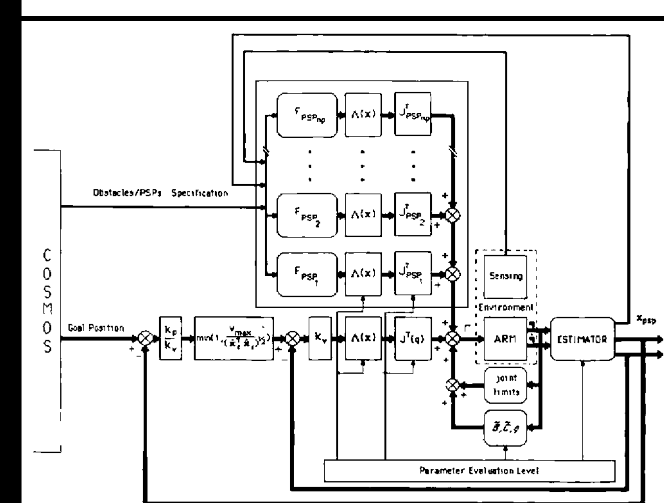
Time-bounded computation

Responsive a dynamic environment
(moving obstacles, goals, new data)

Collision Avoidance

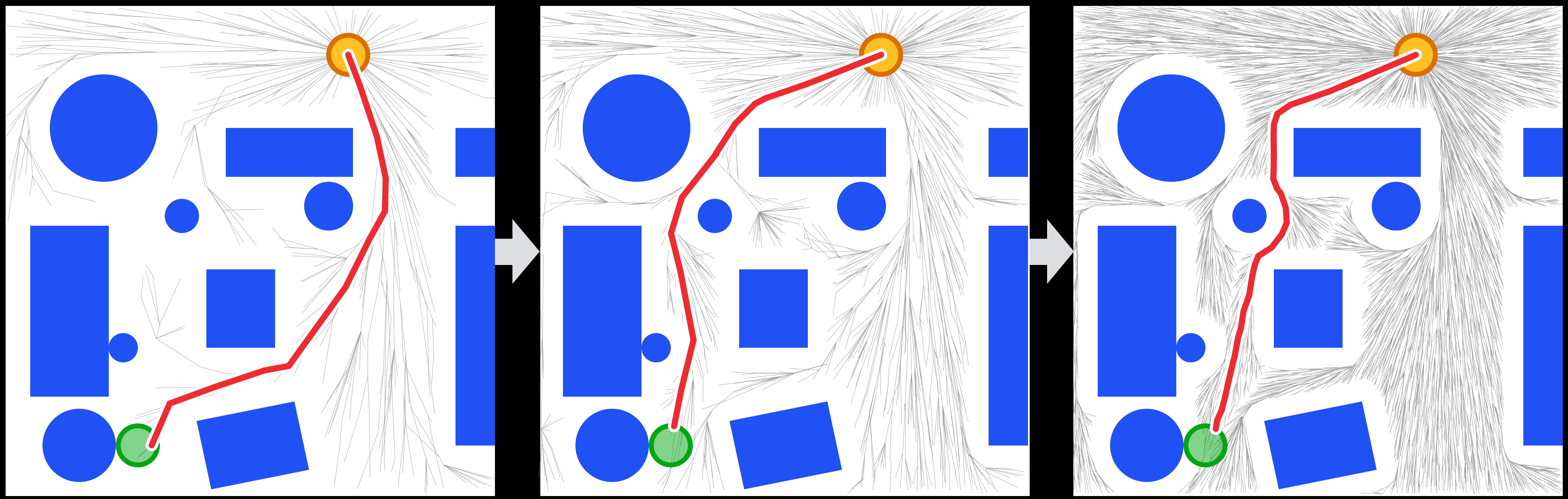
- Reformulation of path planning into collision avoidance.
- Potential fields
- Real-time ✓
- Problem: gets stuck in local minima





[Khatib 1986]

Anytime Planners

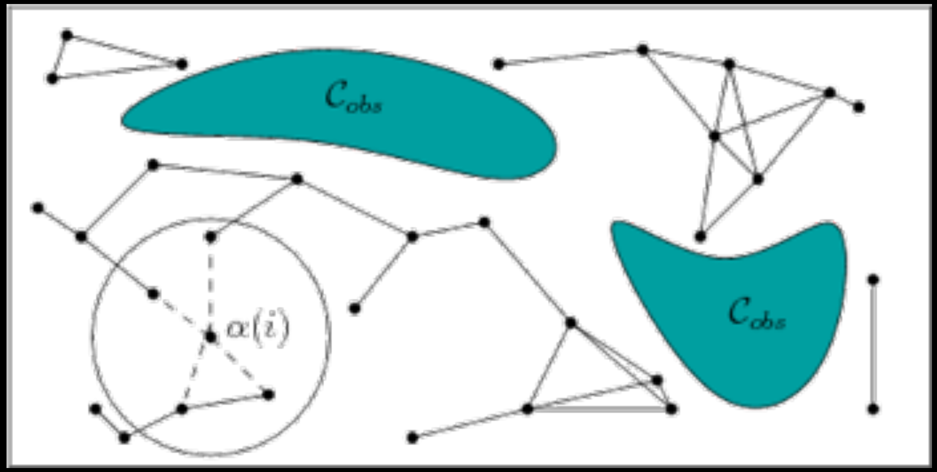


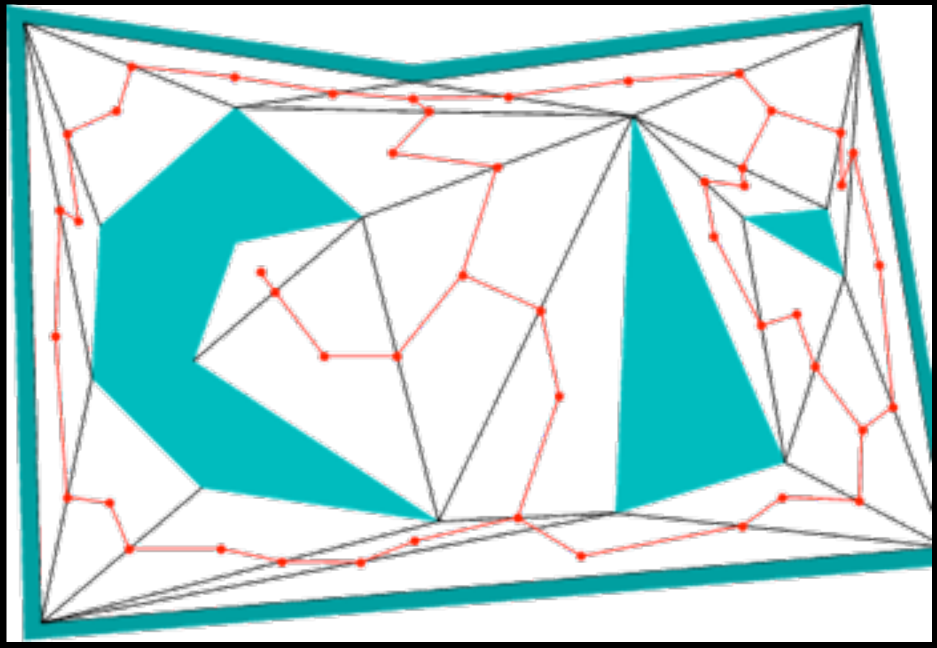
- Incrementally build a planning tree
- May be stopped at **anytime**
- Example: RRT, RRT*
- Real-time?
- Reactivity: rapid re-planning + some luck

[Image source: Ichnowski 2013]

Roadmap Planners

- Pre-computes a roadmap (connectivity of freespace)
- Motion plan = graph search
- Real-time?
- Example: PRM
- Reactivity must come from another task.

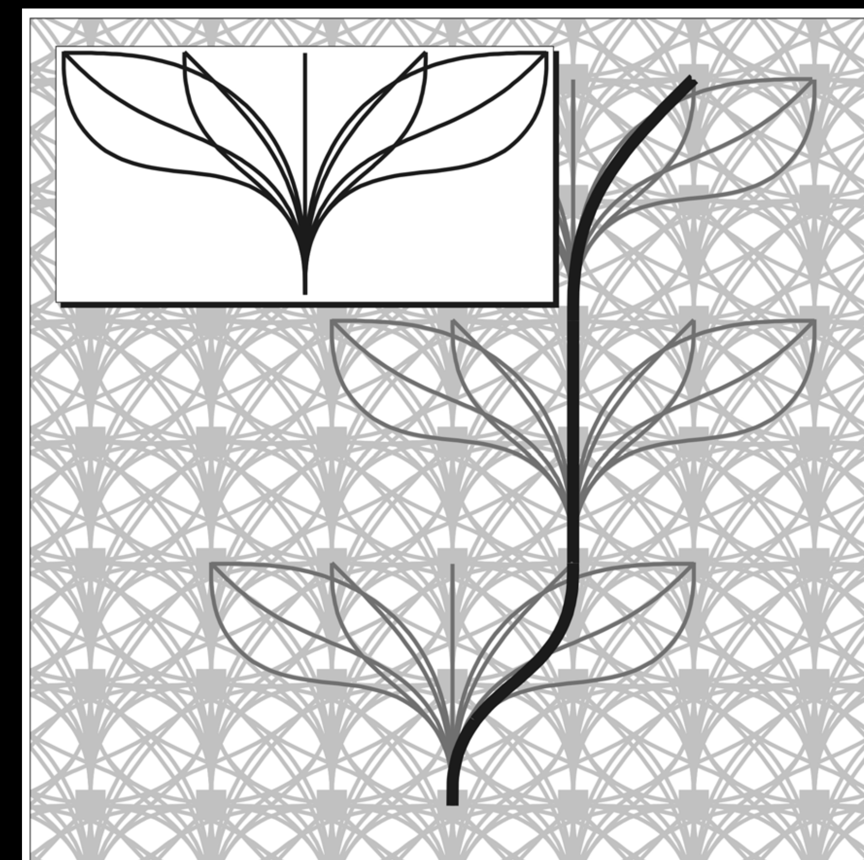




[Image source: LaValle 2006]

Grid/Lattice Planners

1. Discretize space
2. Plan in the discretized space



Often done with A* or variant.

Weighted A*: reduced plan optimality & compute time

D* is reverse A* + keep data for next compute cycle

[Image source: Pivtoraiko 2006]

A*

- Provably optimal
- What if graph changes during the search?
  (e.g., dynamic environment)
- $\mathbf{O(b^d)}$ $\mathbf{d}$ = solution length, $\mathbf{b}$ = branching factor.
  (polynomial if search space is tree*)

Real-Time Heuristic Search

Minimin + α-Pruning

1. Depth-limited horizon search
2. A* metric frontier (S)
   $f(x) = g(x) + h(x)$
3. Take step towards best frontier node
4. repeat.

- Assign $\alpha = \min_{x \in \mathbf{S}} f(x)$
- Prune search when
  $f(x) \geq \alpha$

[Korf 1988]

Real-Time A*

- Use Minimin w/ α-Pruning in “planning mode”
- RTA* used in execution.

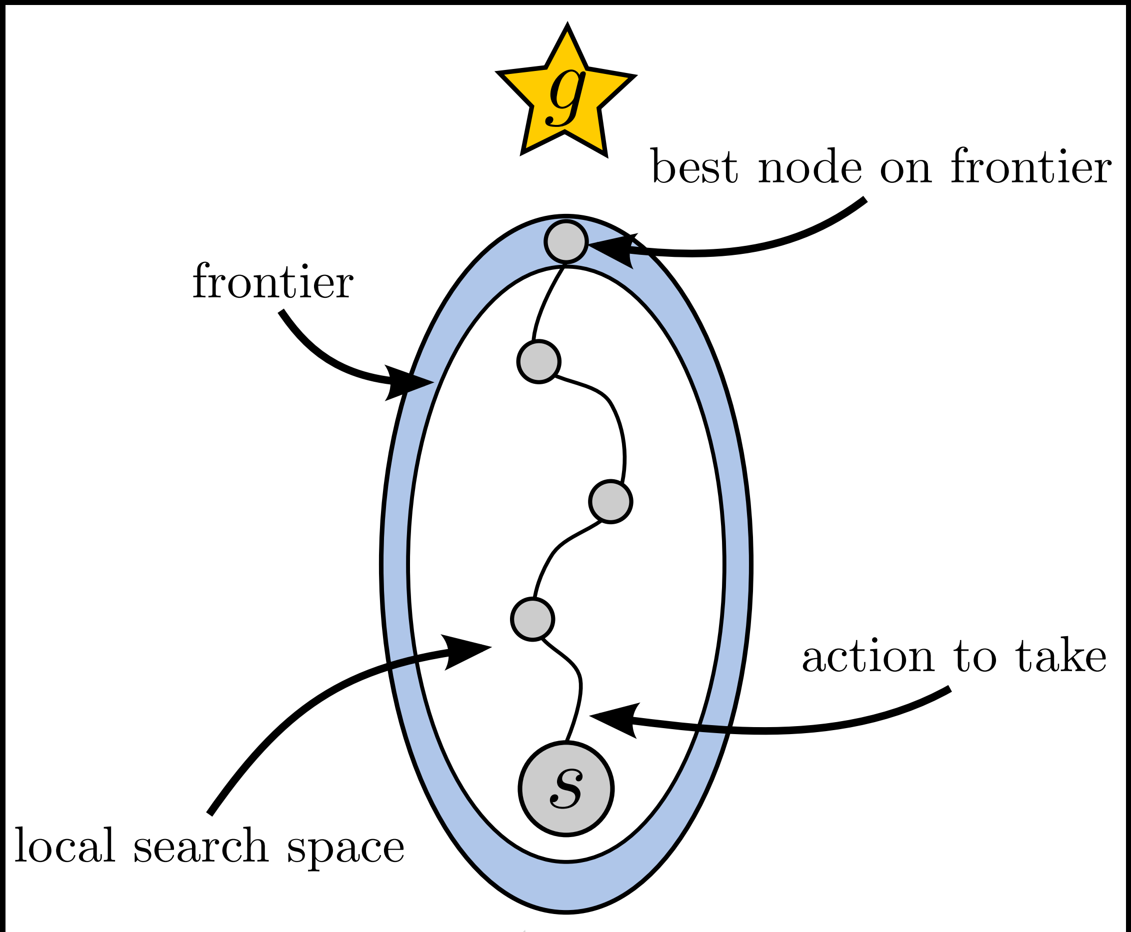
RTA*: At x_i, what is x_{i+1}?

1. Choose $x_{i+1} = \underset{x' \in \text{neighbors of } x_i}{\operatorname{argmin}} g(x') + h(x')$
2. Store ***second*** best $g(x') + h(x')$ for x_i

[Korf 1988]

Partitioned-Learning RTA*

- Start w/ RTA* (depth-limited search)
- Take step towards best path
- "Learn" $\mathbf{h}(\mathbf{x})$ of all frontier
- Split $\mathbf{f}(\mathbf{x})$ into dynamic and static components



$$f(x) = g_s(x) + g_d(x) + h_s(x) + h_d(x)$$

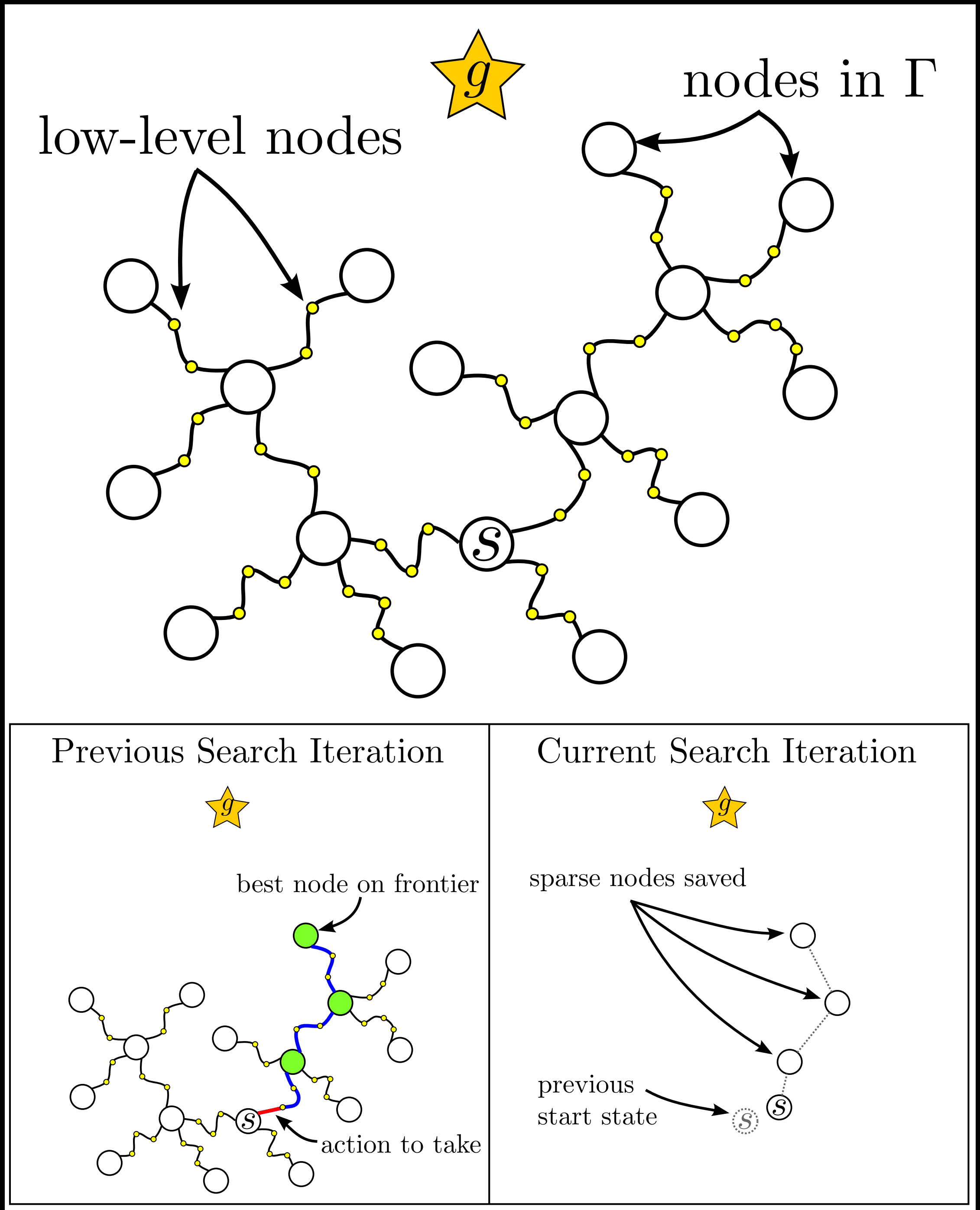
[Cannon et. al. 2014]

Real-Time R* (RTR*)

R* ≈ RRT + A*

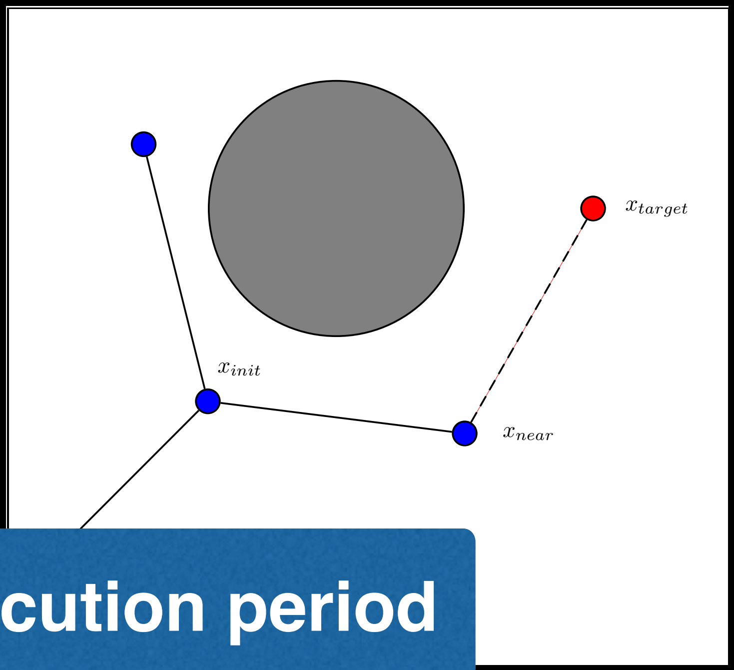
RTR*

- fixed # of node expansions
- choose best frontier node (path and min g(x) + h(x))
- geometric expansion limits for difficult nodes
- path reuse



[Cannon et. al. 2014]

Hard-Real-Time
Rapidly-exploring Randomized Trees




```
procedure HRT_PLANNER
  t_next = current_time()
  loop
    yield until t_next
    t_next = t_next + T_p
    B = updated map
    q_init = current vehicle state
    q_goal = current goal states
    T = BUILD_RRT(q_init, q_goal, n)
    path = EXTRACT_PATH(T)
    publish path
```

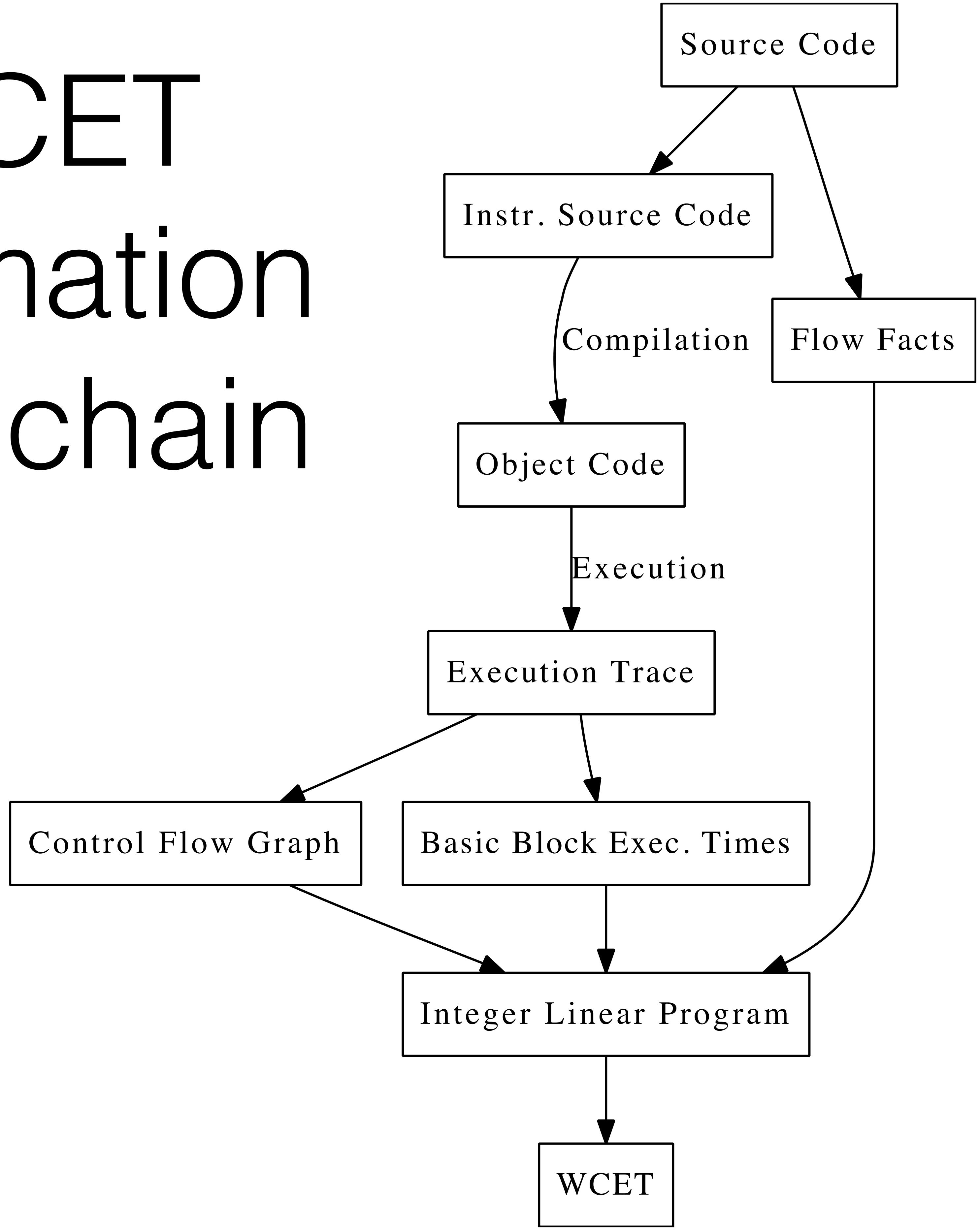
Execution period

solution or "safe" path

number of samples (WCET analysis)

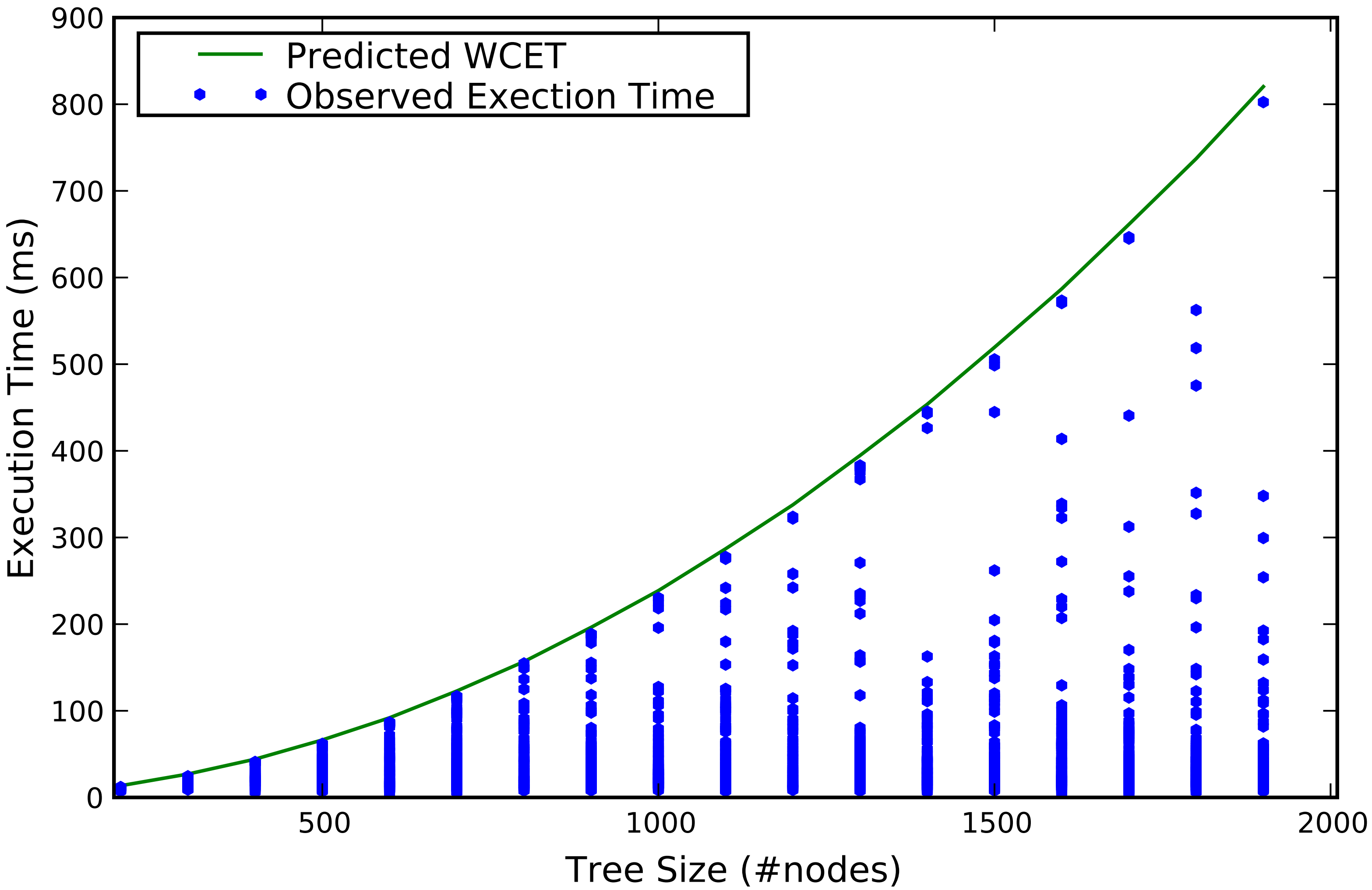
[Walker, 2011]

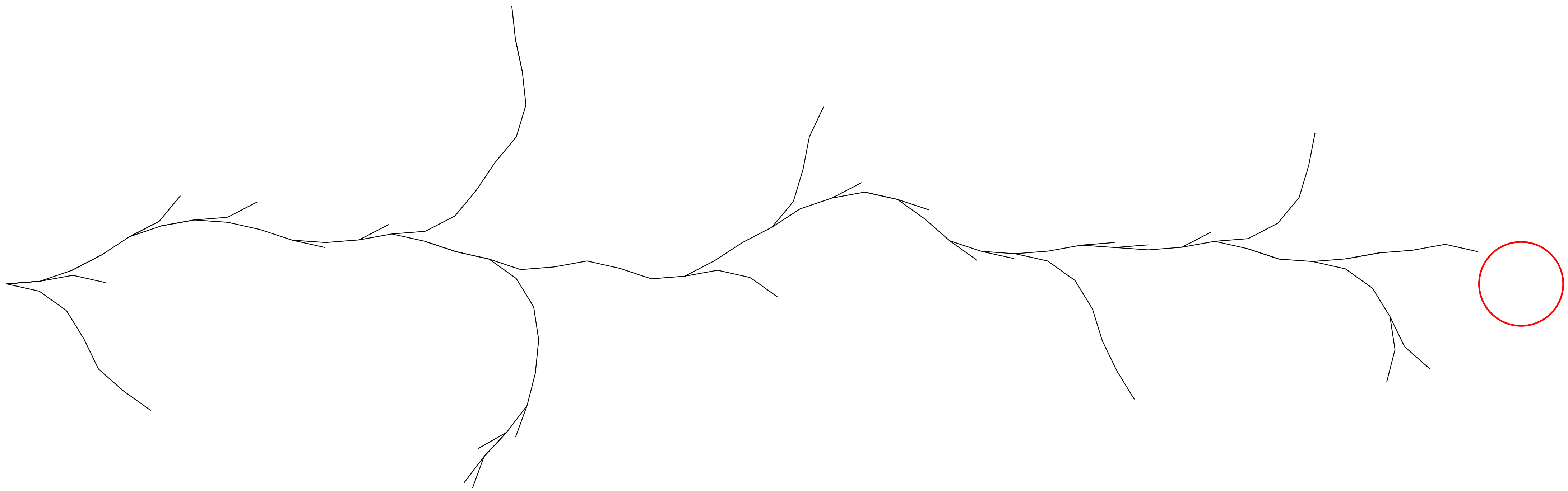
WCET Estimation Tool-chain



[Walker, 2011]

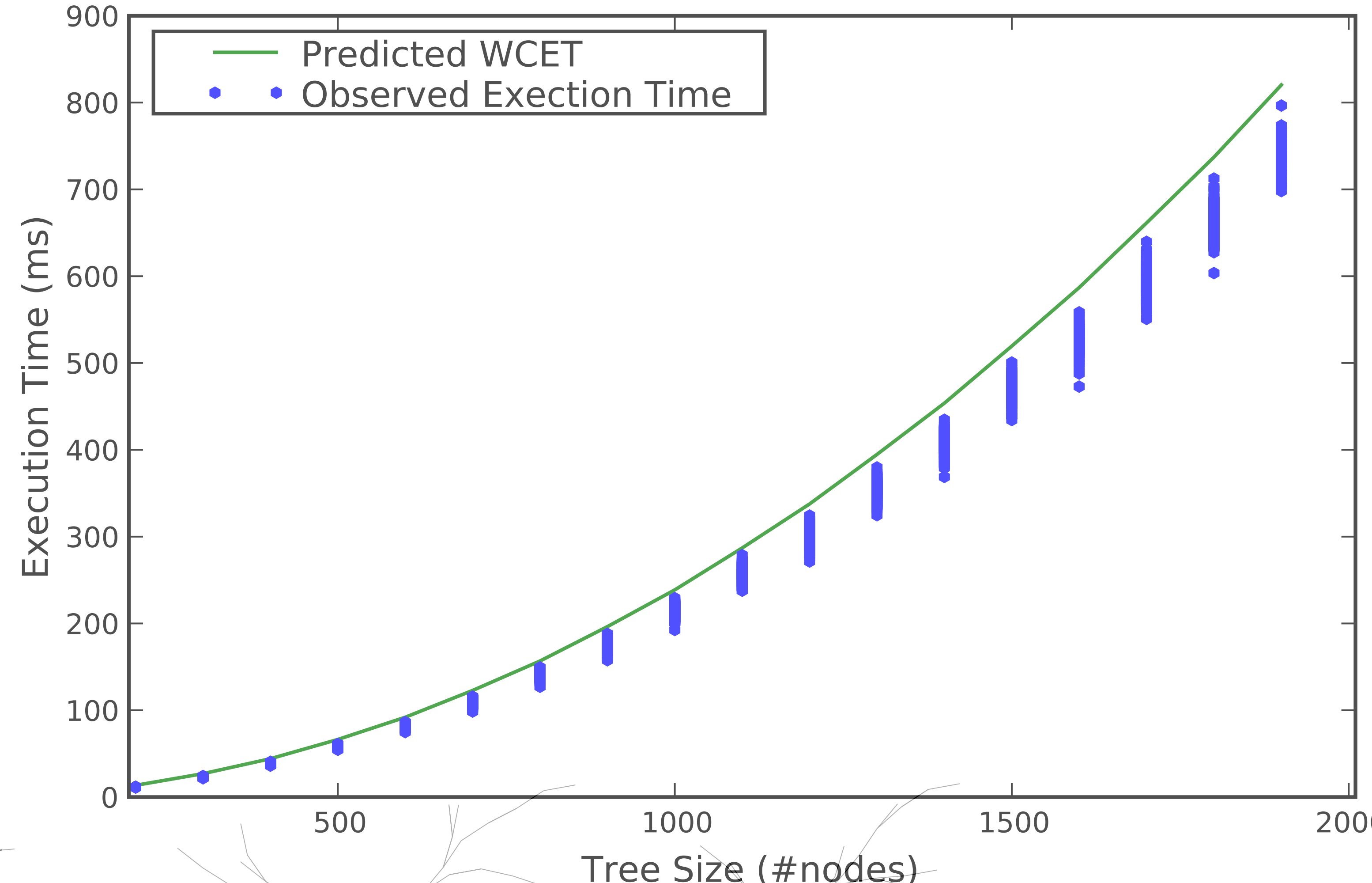
Empty Workspace

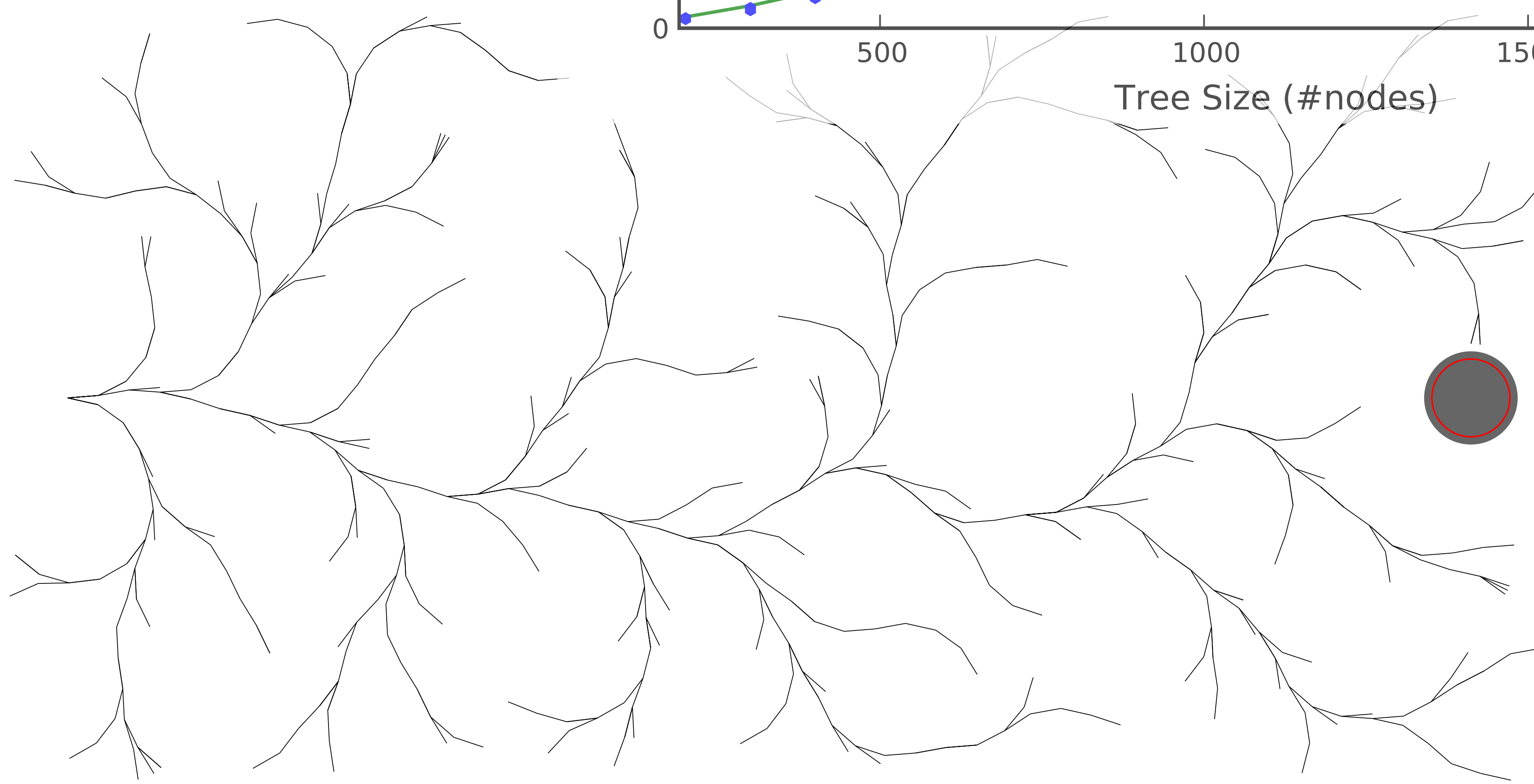




[Walker, 2011]

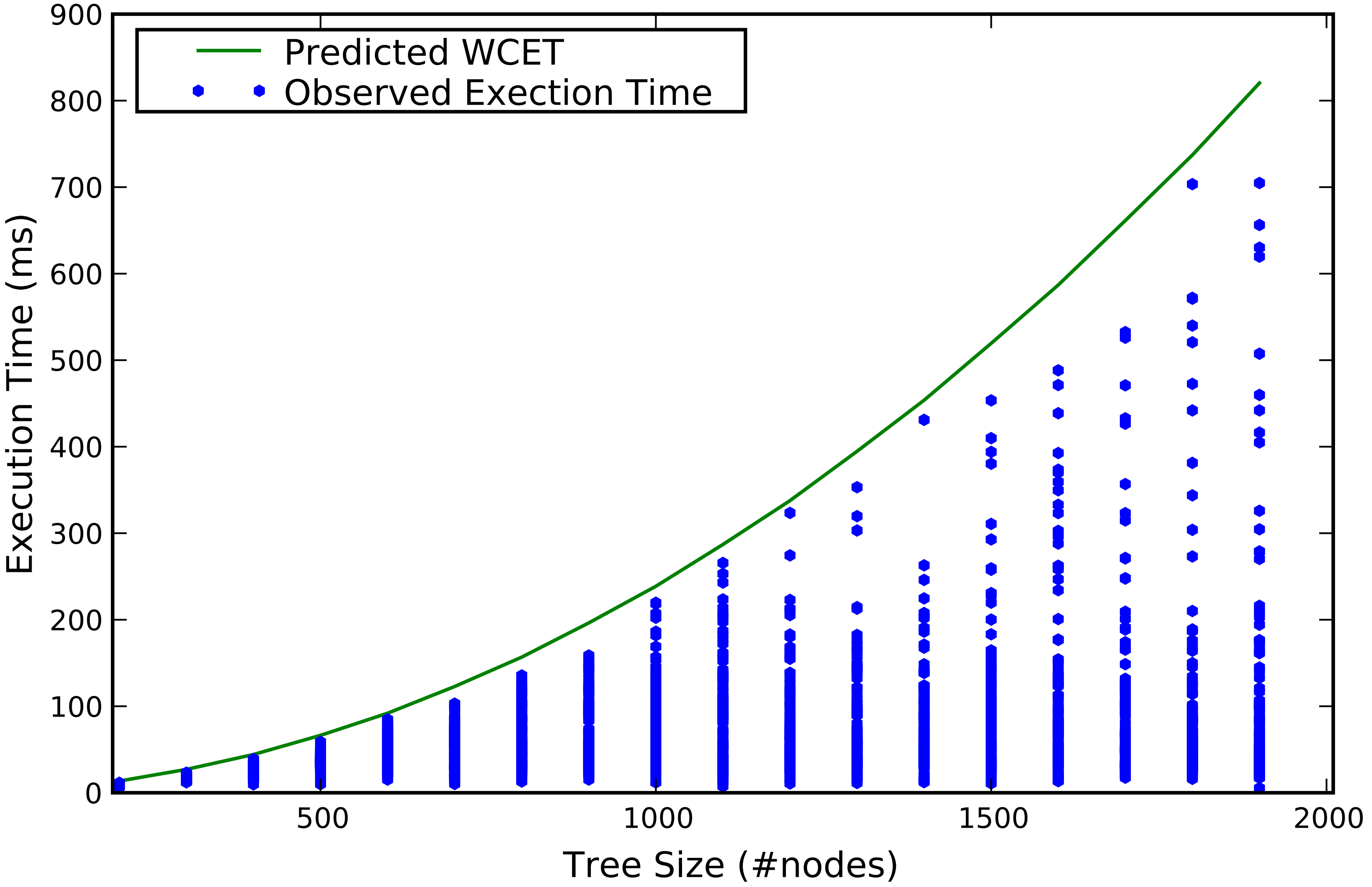
Infeasible Workspace

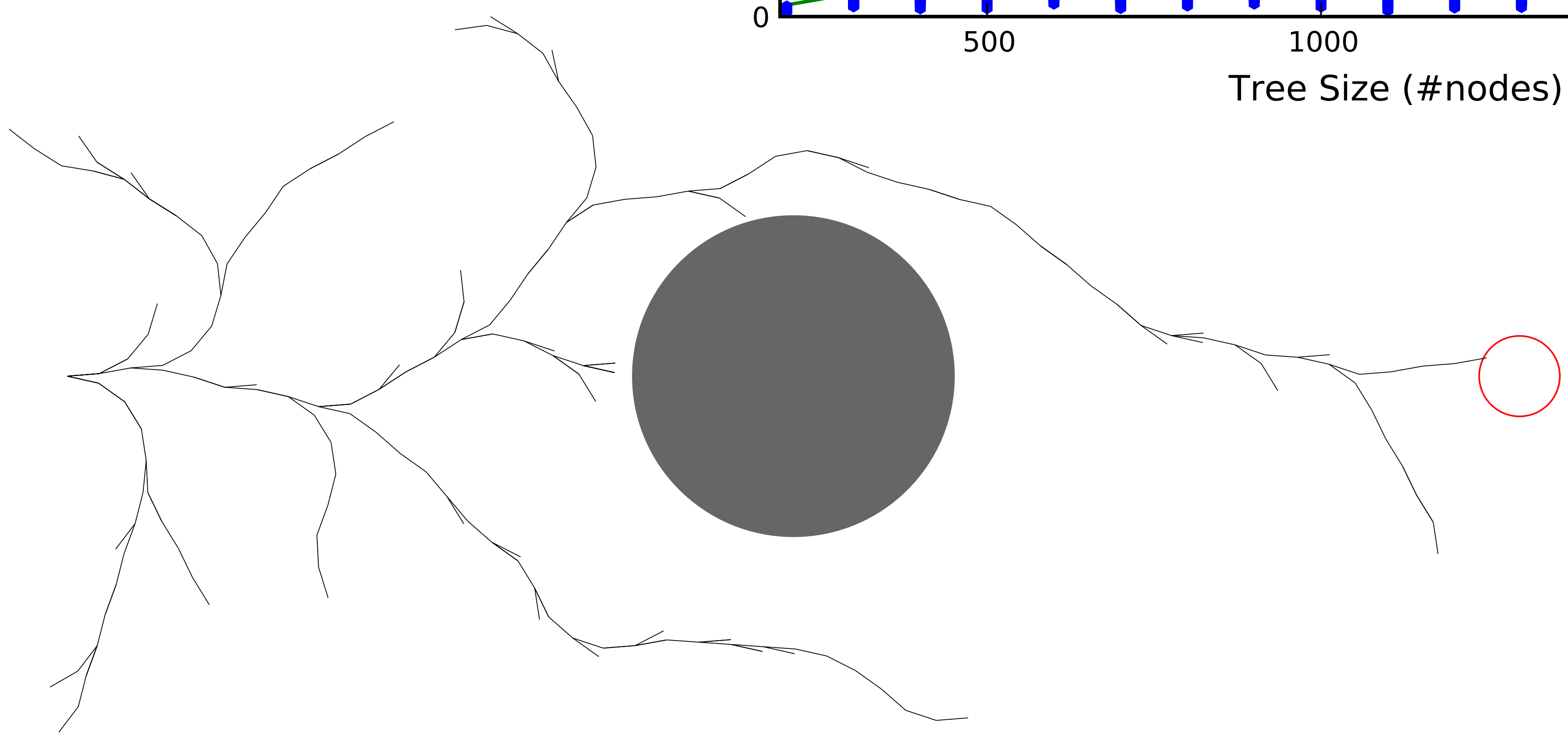




[Walker, 2011]

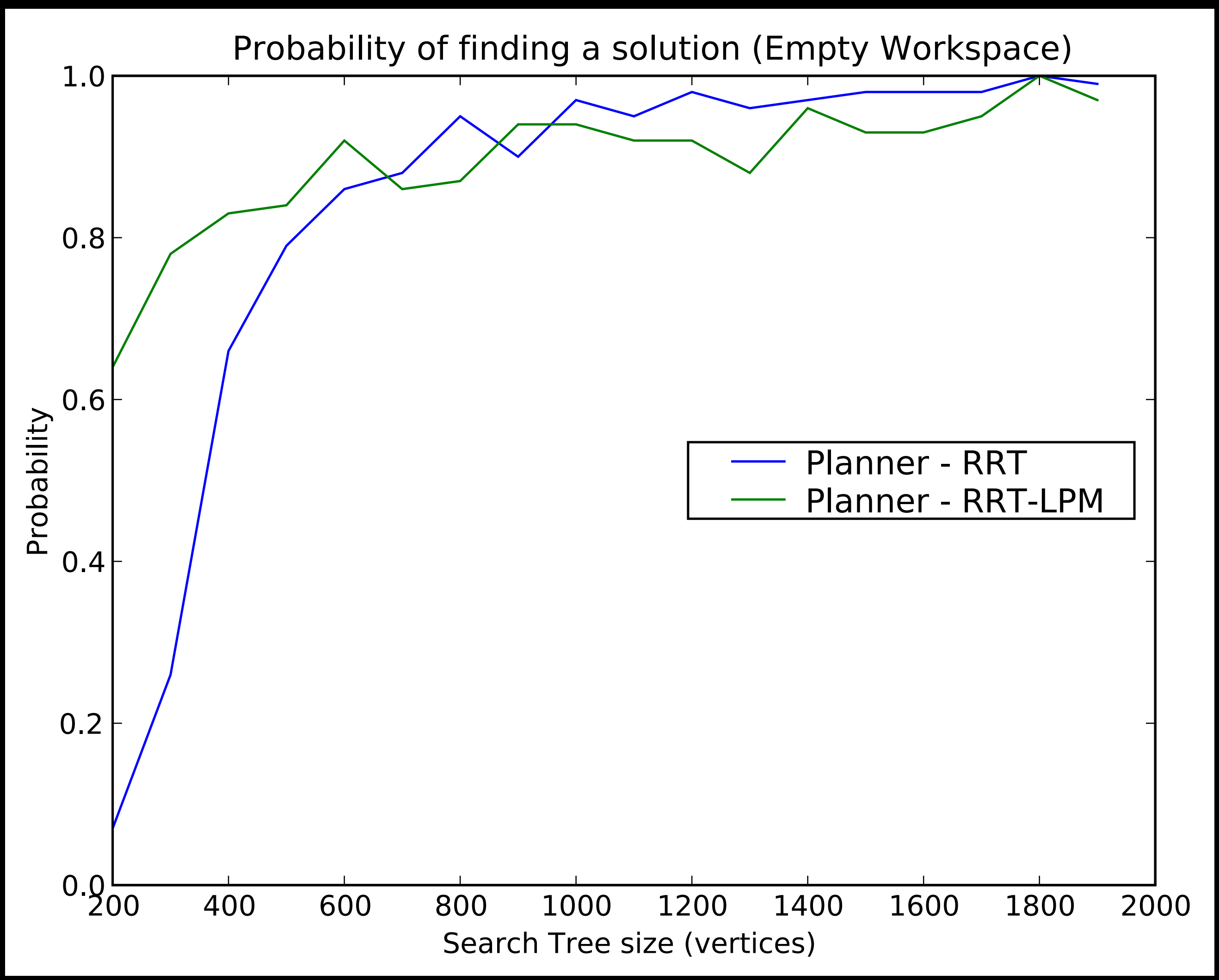
Obstructed Workspace

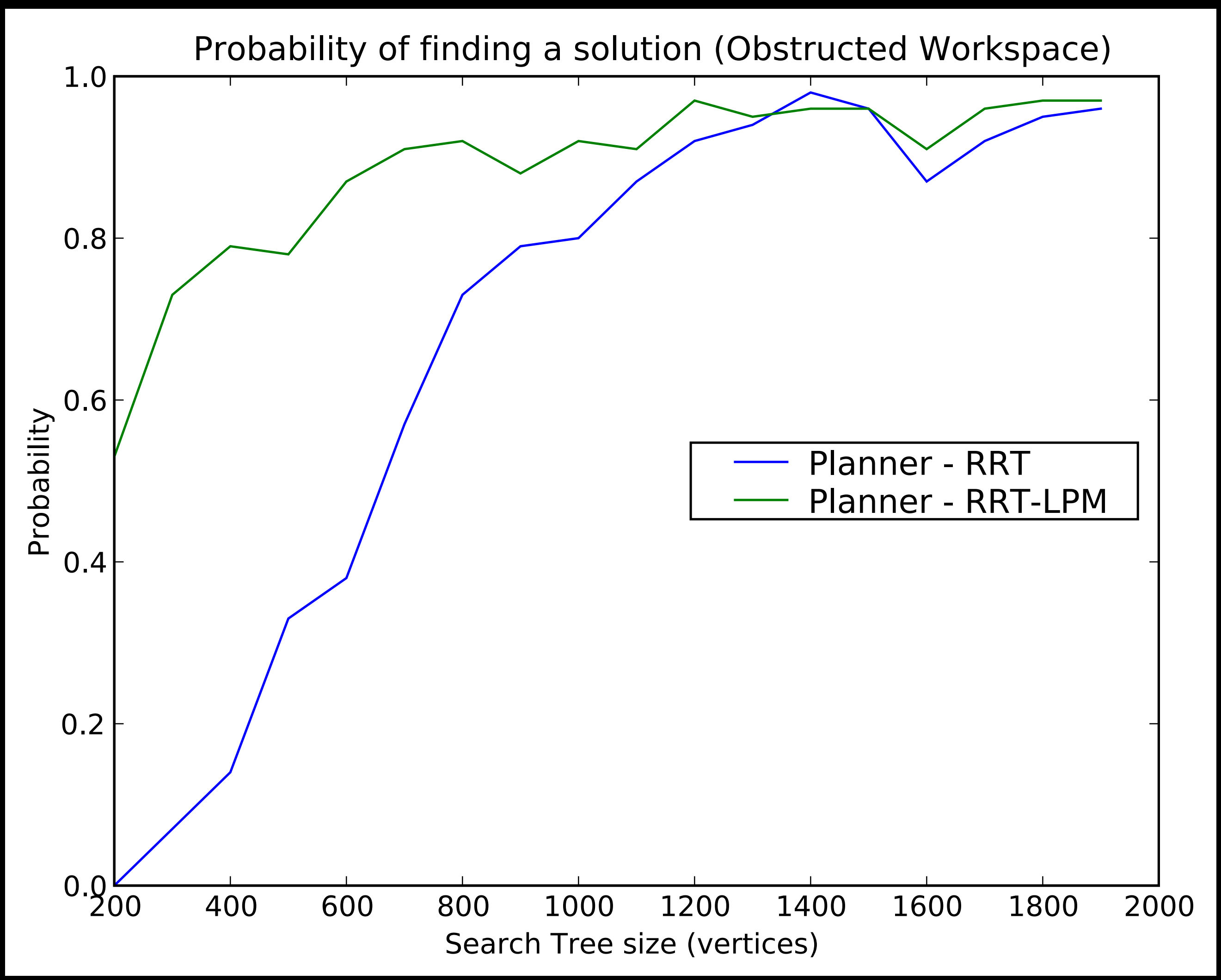




[Walker, 2011]

Solution Probability





[Walker, 2011]

Real-Time Motion Planning for Autonomous Vehicles

Reduce dimensionality of planning problem.
Typically around 4: e.g., (x, y, θ, v)

Discretization and sampling-based approaches

DARPA Urban Challenge 1st place: Tartan Racing (CMU)

local planner at 10 Hz fixed

lattice planner at 10 Hz (nominally)

- Difficult scenarios take “up to a couple of seconds” (motivation for their pre-planning)

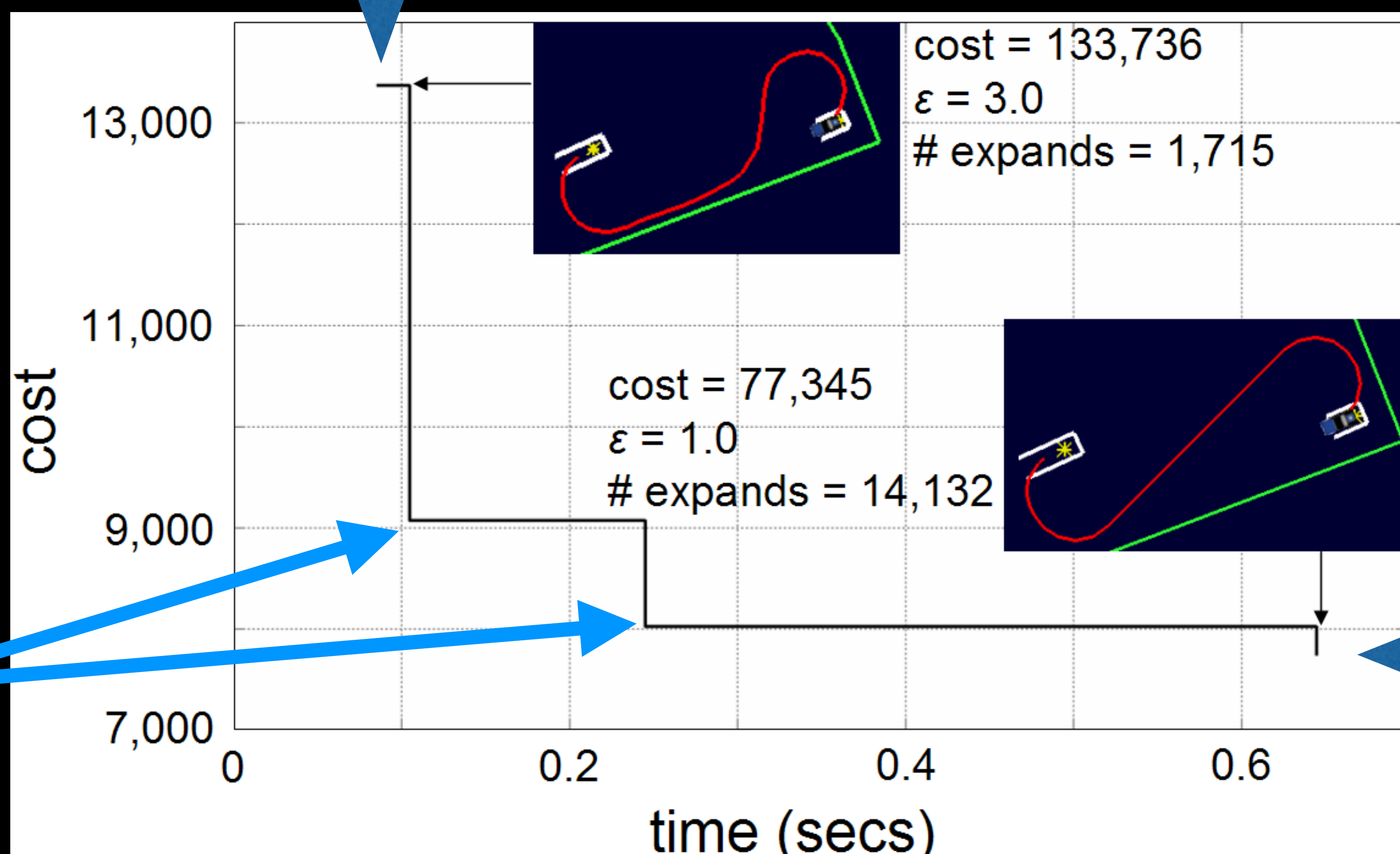
- Anytime planner example:
  first solution in 100 ms, optimal at 650 ms.

- Time for replanning “few ms” for small adjustments to “few seconds” for drastically different trajectories

[Likhachev 2008] & [Ferguson 2008]

DARPA Urban Challenge 1st place:
Tartan Racing (CMU)

solution found
<100 ms



solution improved

optimal solution
< 650 ms

Anytime planner behavior

[Likhachev, et. al. 2008]

DARPA Urban Challenge 1st place:
Tartan Racing (CMU)

Effect of heuristic on A* search

	States Expanded	**Time (seconds)**
h	2,019	0.06
h_{2D}	26,108	1.30
h_{fsh}	124,794	3.49

Implies much higher WCET

[Likhachev, et. al. 2008]

DARPA Urban Challenge 1st place:

Tartan Racing (CMU)

"One of the important lessons learned during the development of this system was that it is often extremely beneficial to exploit prior, offline processing to provide efficient online planning performance."

– Ferguson, Howard, and Likhachev

DARPA Urban Challenge 2nd place:
Stanford Racing

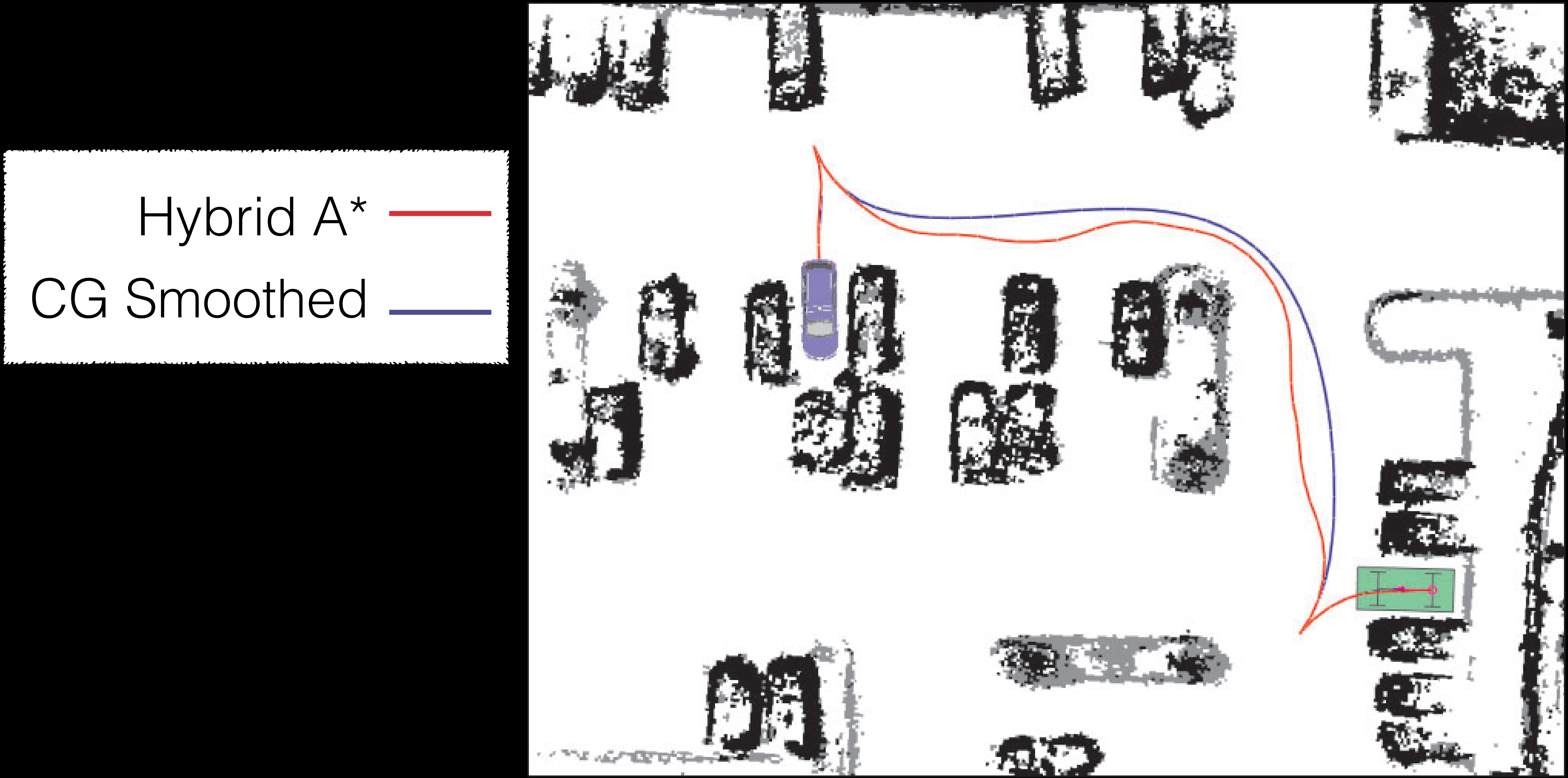
- Sensors at 10 Hz
- RNDF[1] editor at 10 Hz
- Full replanning: 50 to 300 ms
  1. hybrid A* (unnatural swerves)
  2. conjugate-gradient descent smooth (0.5 m)
  3. interpolation (5 to 10 cm)

Grid: 160 m x 160 m x 360°
Resolution of 1 m x 1 m x 5°

[1] route network definitions file

[Montemerlo et. al. 2009] [Dolgov et. al. 2010]

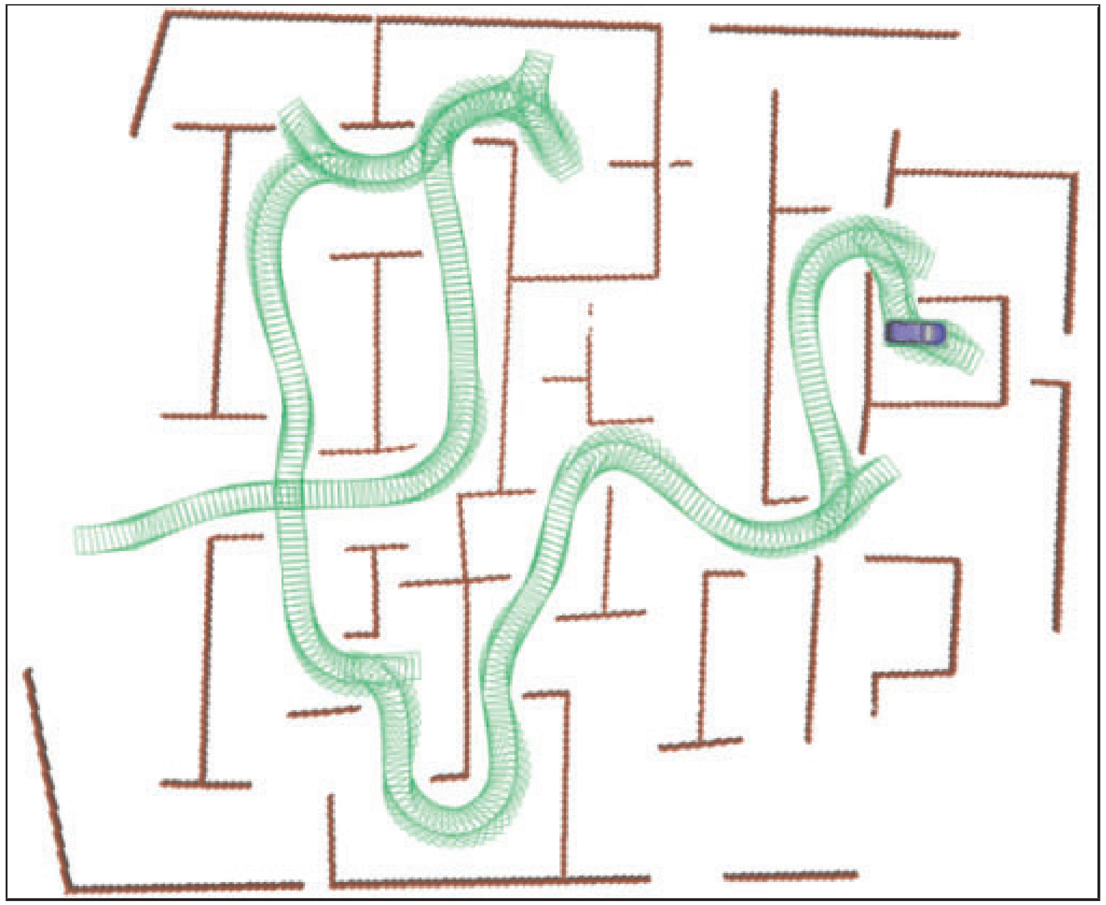
DARPA Urban Challenge 2nd place:
Stanford Racing

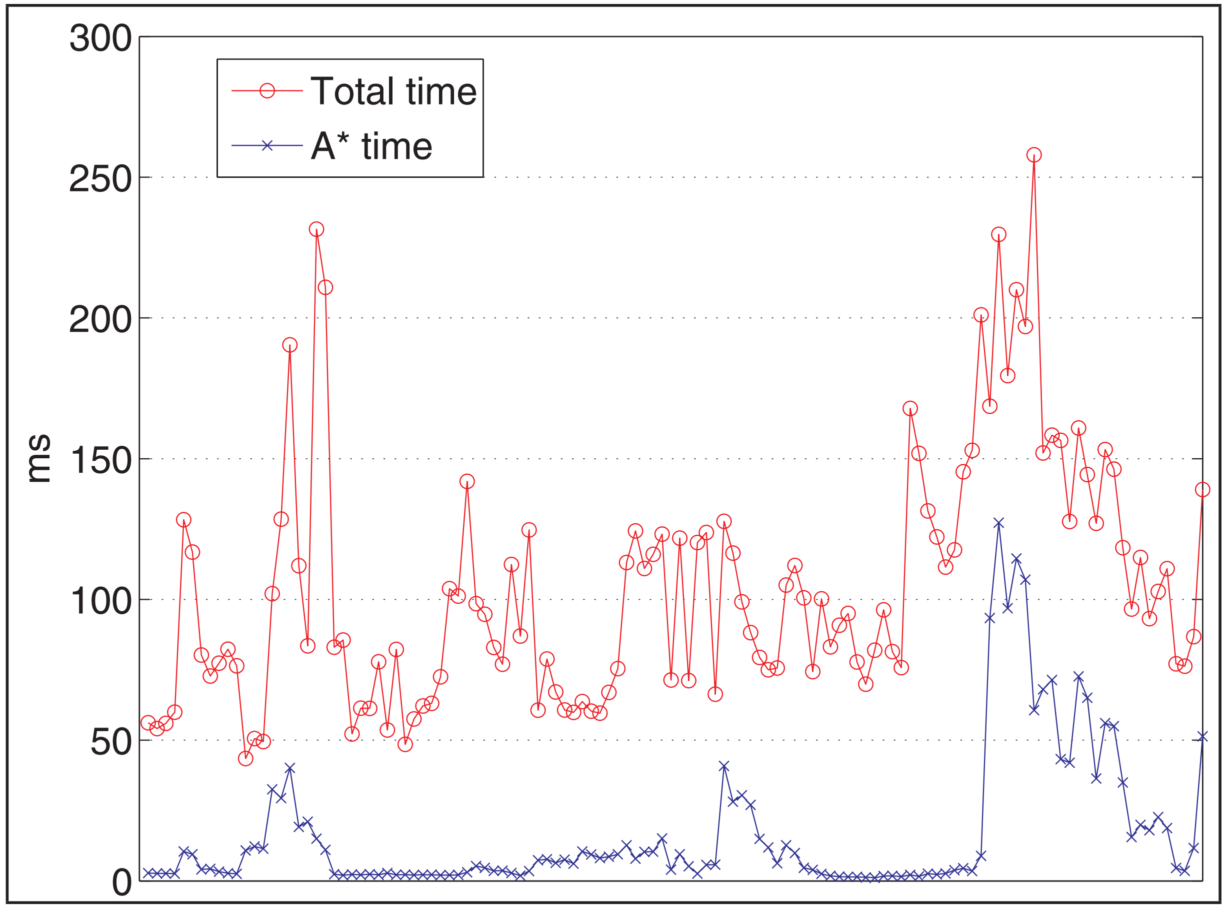


[Dolgov et. al. 2010]

DARPA Urban Challenge 2nd place:

Stanford Racing





[Dolgov et. al. 2010]

DARPA Urban Challenge 4th place: MIT

- Drivability map updated 10 Hz
- Controller ran at 25 Hz
- RRT at 10 Hz
  - 700 samples per second

[Kuwata, et. al. 2009]

DARPA Urban Challenge 4th place: MIT

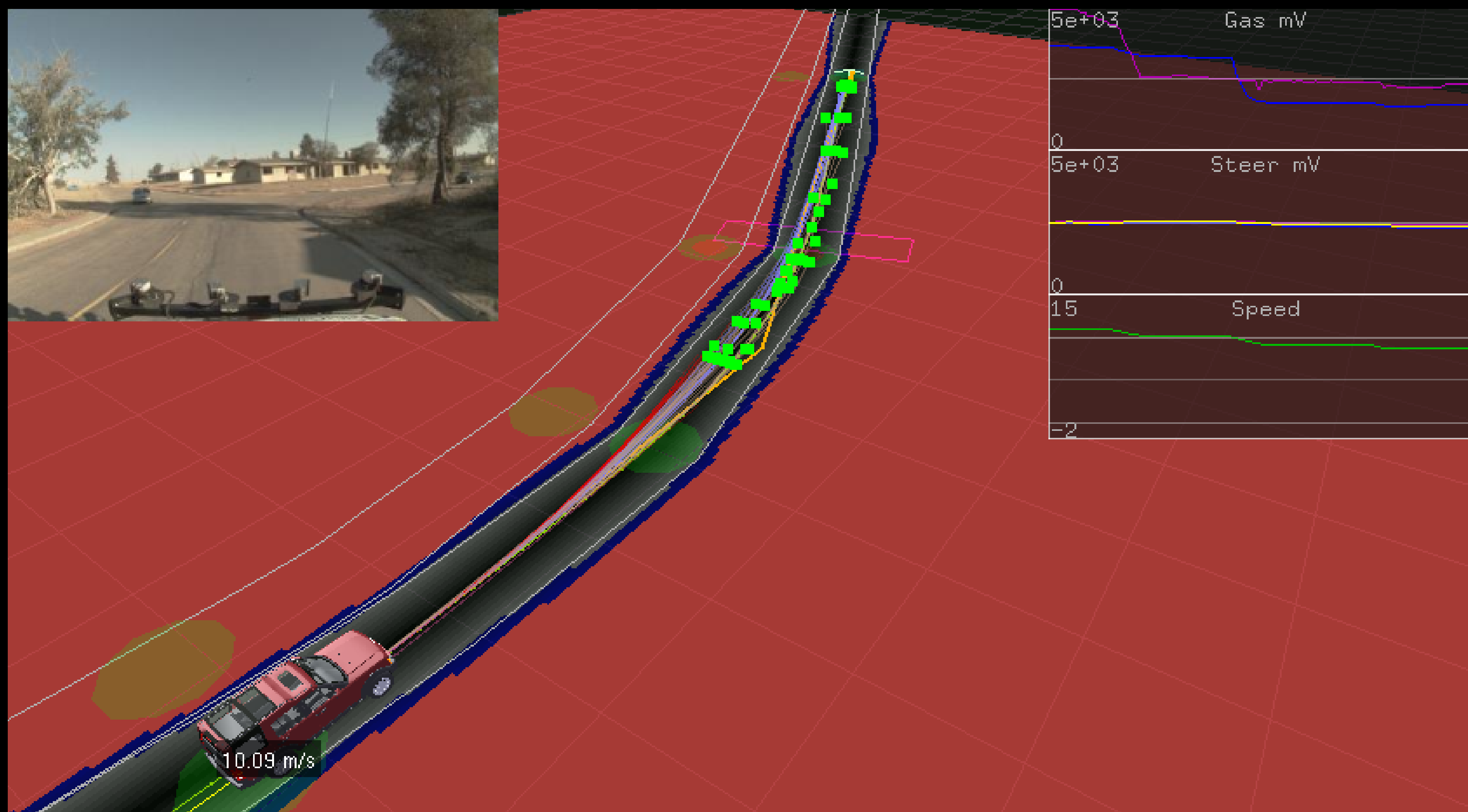
```
procedure RRT_execution_loop
repeat
  update vehicle states and env
  while (t < t_0 + Δt)
    EXPAND_RRT_TREE()
  repeat {
    τ = EXTRACT_BEST_SAFE_PATH()
    if NO safe path
      E-STOP! & restart
  } until (clear(x) ∀x ∈ τ)
  send τ to controller
```

hard real-time constraint

"take appropriate action"

[Kuwata, et. al. 2009]

DARPA Urban Challenge 4th place: MIT



Lane following on a curve at 22.4 mph.
The green dots are safe stopping nodes.

[Kuwata, et. al. 2009]

DARPA Urban Challenge finalist: AnnieWay (KIT)

- DNF. Froze at entrance to traffic circle (who doesn't their first time?)

  Software exception during mode switch
  Caught by error handler, and left hanging
  Not observable by watchdog module.

- One of the few cars that drove collision-free

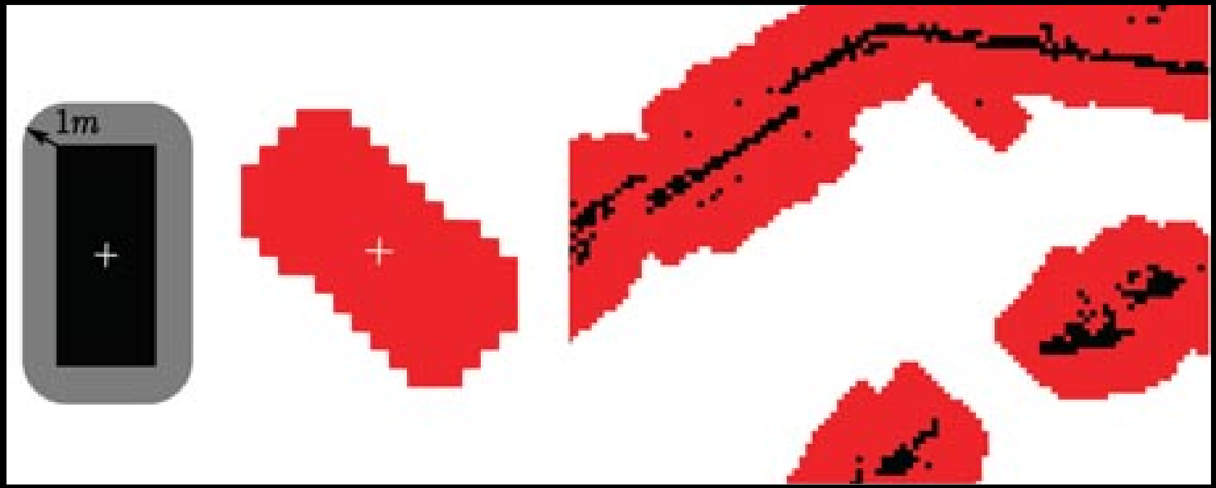
- One of the authors Matthias Goebl from Institute for Real-Time Computer Systems, Technical University of Munich

[Kammel, et. al. 2008]

DARPA Urban Challenge finalist: AnnieWay (KIT)

- Multi-level control:
  A. Mission planning
  B. Maneuver planning
  C. Collision avoidance
  D. Reactive layer
  E. Vehicle control

RT*



Car and Kernel w/ 1m safety buffer

Environment w/ car convolution

- Motion planning on discretized grid of 3D configuration space using A*.

- Convolutional filters used to precompute free C-space.

[Kammel, et. al. 2008]

Conclusions

- Real-time motion planning difficult
- No guarantees on solution
- Multiple levels of planning
- Time-bounded computation
- Generate “safe routes”
- Keep around information between task cycles

Thank you