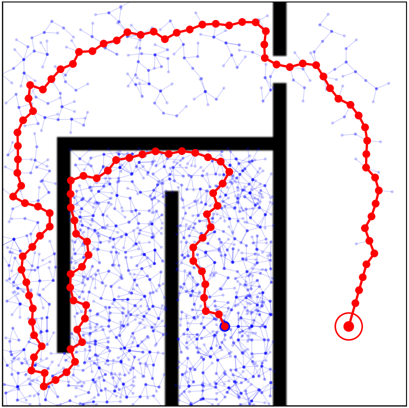
Improving Path Planning Methods Using Machine Learning

Artyom Tsoy

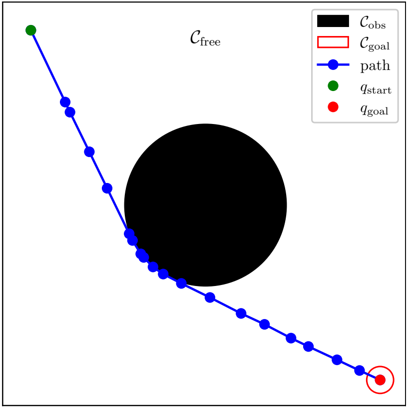
Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

supervisor: Ing. Vojtěch Vonásek, Ph.D.

12.06.2024



(a) RRT.



(b) 2D workspace with a 2D $\mathcal{C}$-space.

Figure: RRT and a worspace example.

Motivation

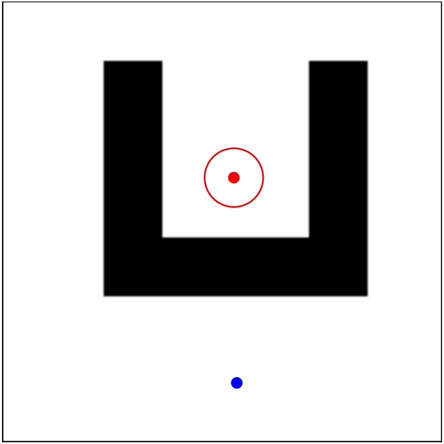


Figure: An example of the 2D workspace with a 2D $\mathcal{C}$-space.

- Efficient exploration of the $\mathcal{C}$-space;
- Sampling points in $\mathcal{C}_{\text{free}}$;
- Real-time learning;
- Finding better solutions with the same number of iterations.
- Inspired by the work of [1].

Proposed solution

$$\hat{f}_{\mathcal{X}}(x) = \frac{1}{m} \sum_{i=1}^{m} K\left(x - x_i\right)$$

$$K(x) = (2\pi)^{-\frac{d}{2}} \cdot \det(\mathbf{H})^{-\frac{1}{2}} \cdot e^{-\frac{1}{2}x^T H^{-1} x}$$

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$$

$$P(x|y=1) = \hat{f}_{\mathcal{X}_{\text{obs}}}(x) = \frac{1}{|\mathcal{X}_{\text{obs}}|} \sum_{x' \in \mathcal{X}_{\text{obs}}} K\left(x - x'\right)$$

$$P(x|y=0) = \hat{f}_{\mathcal{X}_{\text{free}}}(x) = \frac{1}{|\mathcal{X}_{\text{free}}|} \sum_{x' \in \mathcal{X}_{\text{free}}} K\left(x - x'\right)$$

$$P(y=1) = \frac{|\mathcal{X}_{\text{obs}}|}{|\mathcal{X}_{\text{obs}}| + |\mathcal{X}_{\text{free}}|}; \quad P(y=0) = \frac{|\mathcal{X}_{\text{free}}|}{|\mathcal{X}_{\text{obs}}| + |\mathcal{X}_{\text{free}}|}$$

Proposed solution

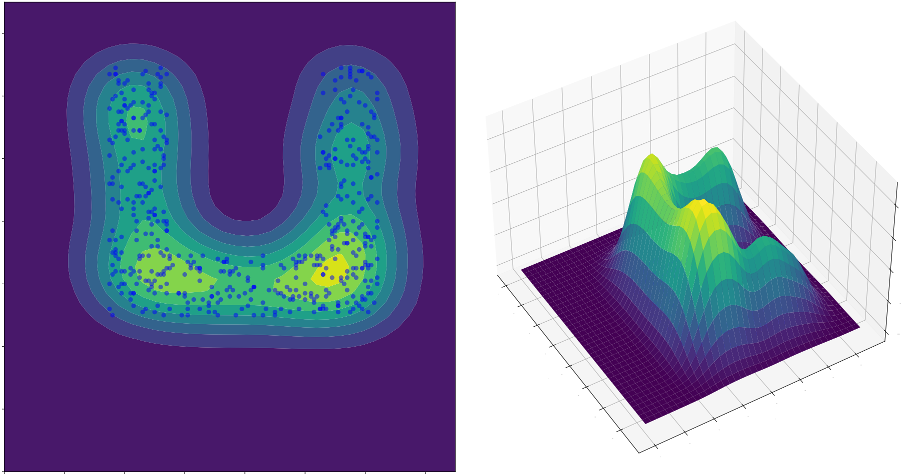


Figure: Obstacle density.

Proposed solution

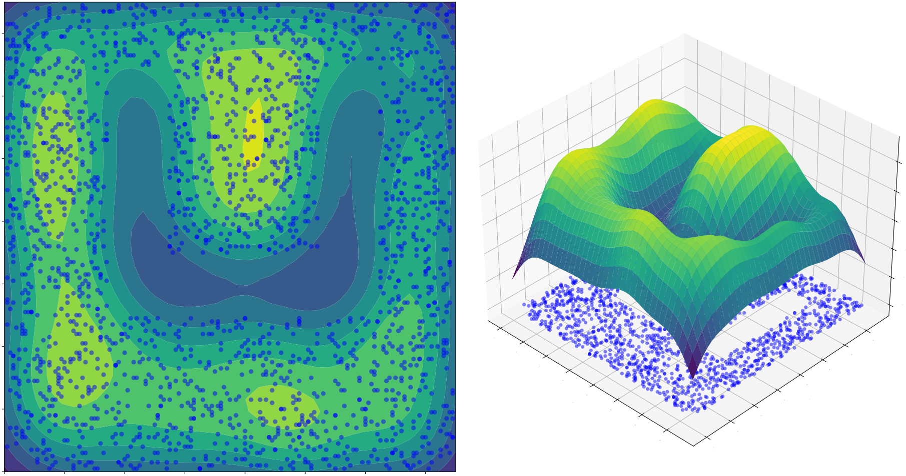


Figure: Obstacle-free density.

Proposed Solution

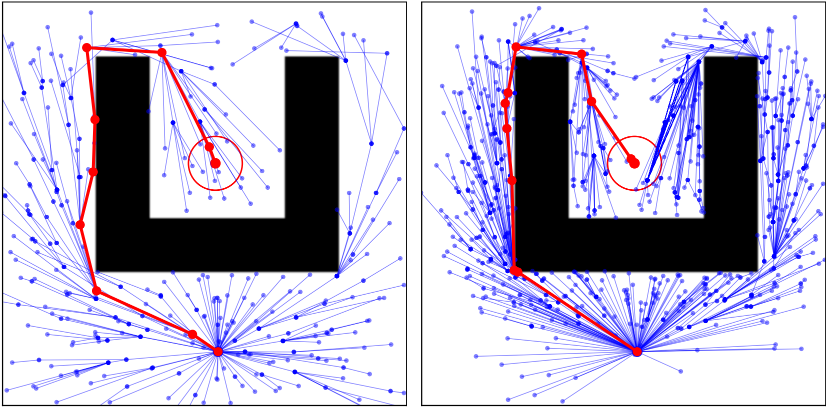
Algorithm 1: Sample Density

Data: $\mathcal{X}_{\text{obs}}$, $\mathcal{X}_{\text{free}}$
Result: Predicted sampled point x

$\gamma_{\text{free}} \leftarrow 0$;
$\gamma_{\text{obs}} \leftarrow 1$;
while $\gamma_{free} < \gamma_{obs}$ **do**
  $x_{\text{rand}} \leftarrow$ RandomConfiguration();
  $P_{\text{free}} \leftarrow \frac{|\mathcal{X}_{\text{free}}|}{|\mathcal{X}_{\text{free}}|+|\mathcal{X}_{\text{obs}}|}$;
  $P_{\text{obs}} \leftarrow 1 - P_{\text{free}}$;
  $b_{\text{free}} \leftarrow$ DensityEstimator(x_{rand}, $\mathcal{X}_{\text{free}}$);
  $b_{\text{obs}} \leftarrow$ DensityEstimator(x_{rand}, $\mathcal{X}_{\text{obs}}$);
  $\gamma_{\text{free}} \leftarrow b_{\text{free}} \cdot P_{\text{free}}$;
  $\gamma_{\text{obs}} \leftarrow b_{\text{obs}} \cdot P_{\text{obs}}$;

$x \leftarrow x_{\text{rand}}$;
return x;

Proposed solution



(a) RRT* (216 u.d).   (b) Improved RRT* (206 u.d).

Figure: RRT* and Improved RRT* comparison.

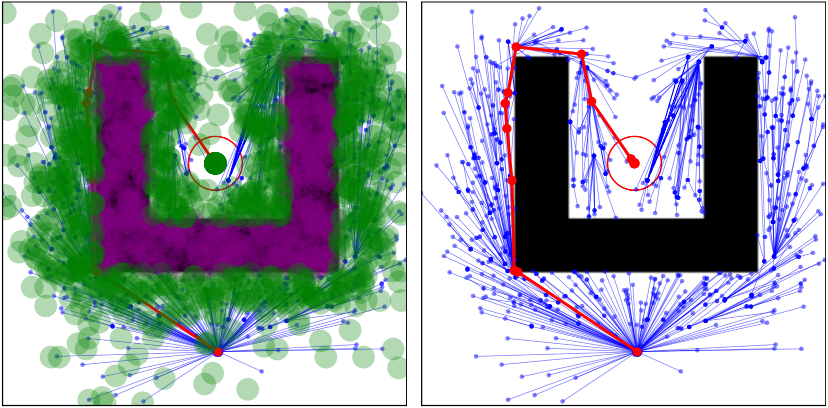
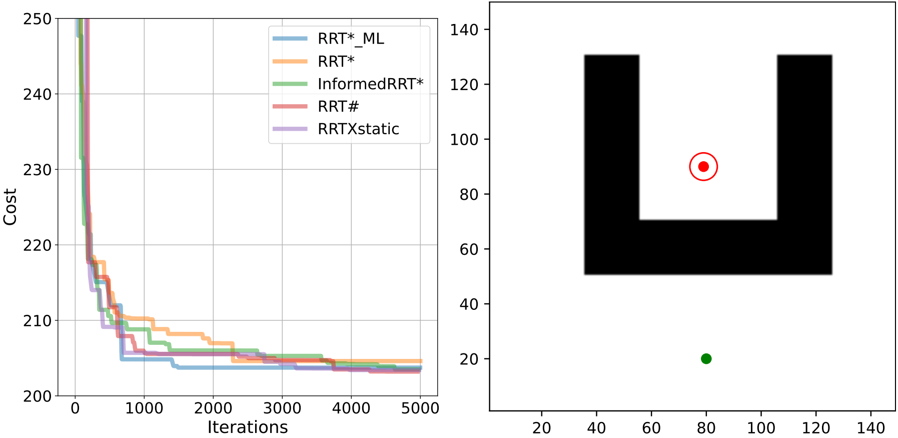


Figure: Learning the $\mathcal{C}$-space.

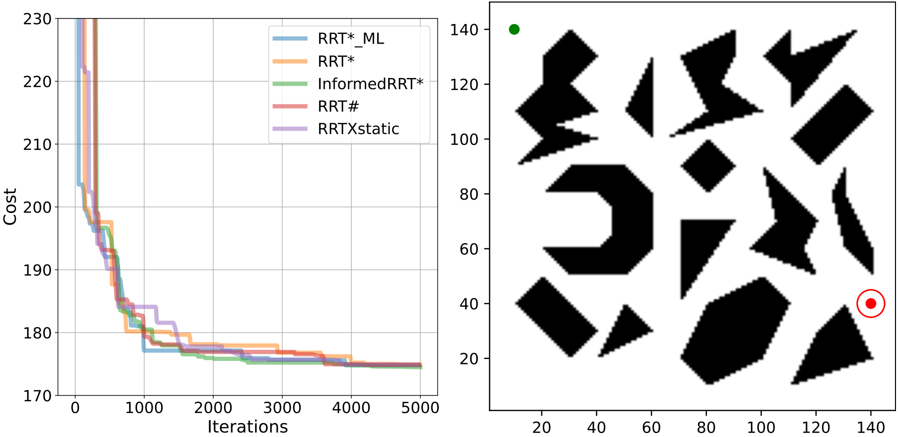
Results



(a) Graph. (b) Map.

Figure: Convergence graph for the corresponding map.

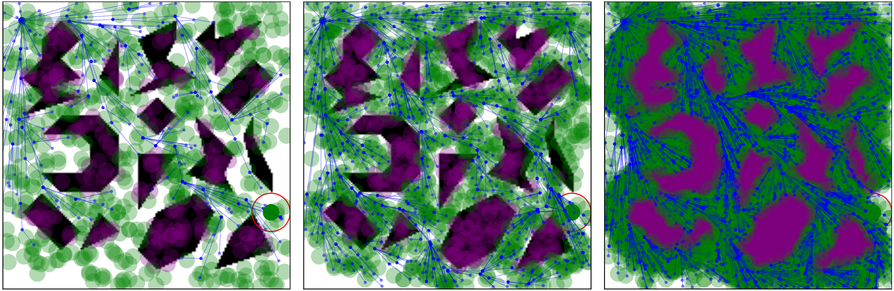
Results



(a) Graph. (b) Map.

Figure: Convergence graph for the cluttered map.

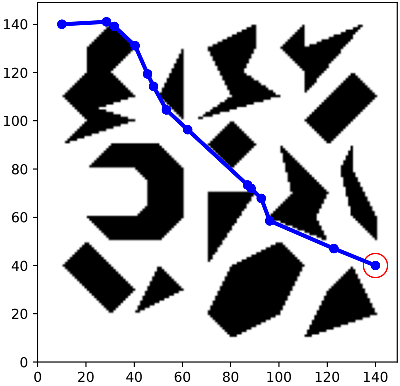
Results



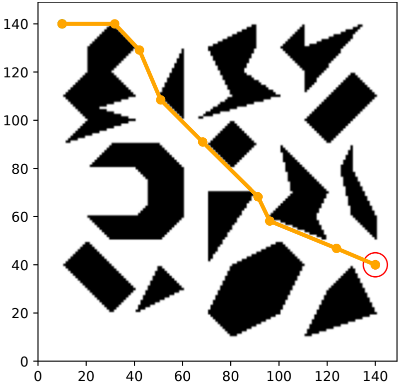
(a) 500 iterations. (b) 1000 iterations. (c) 5000 iterations.

Figure: Learning $\mathcal{C}$-space in the cluttered map.

Results



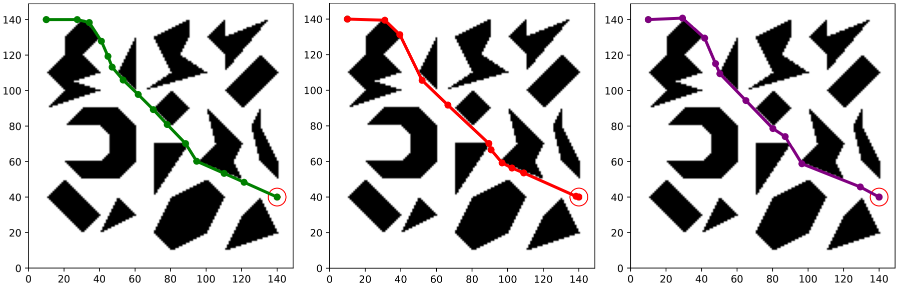
(a) RRT*_ML.



(b) RRT*.

Figure: Found solutions.

Results



(a) Informed RRT*. (b) $RRT^{\#}$. (c) RRT^{X} static.

Figure: Found solutions.

Results

Runtime		
Planner	**Initial solution [s]**	**Final solution [s]**
RRT*_ML	0.278594 ± 0.157754	289.625891 ± 2.611068
RRT*	0.006518 ± 0.001462	0.321272 ± 0.014779
Informed RRT*	0.007330 ± 0.000536	0.259046 ± 0.003154
RRT$^{\#}$	0.011917 ± 0.002803	0.305096 ± 0.010232
RRT^{X} static	0.011775 ± 0.002566	0.305259 ± 0.008140

Table: Time spent to find the first and final solutions in the cluttered map.

Results

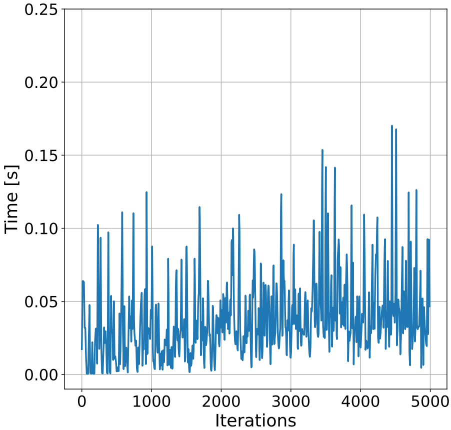


Figure: Time taken to learn predicting points in the 2D workspace with the 2D $\mathcal{C}$-space to be in the C_{free}.

- Implementation in Python;
- Time required to predict points;
- Increasing dataset sizes necessitating the computation of more kernels.

Results

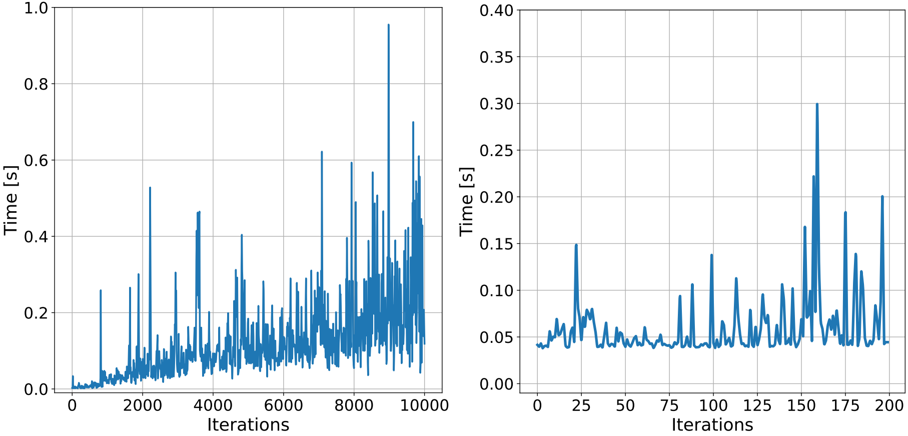


Figure: Time taken to learn predicting points in the 2D workspace with the 3D $\mathcal{C}$-space and in the 3D workspace with the 6D $\mathcal{C}$-space to be in the C_{free}.

Questions from the Opponent

- Can you define “optimal solution”?
- Is RRT* method returning optimal solution?

Thanks for your attention!

Oktay Arslan and Panagiotis Tsiotras.
Machine learning guided exploration for sampling-based motion planning algorithms.
In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2646–2652, 2015.