Euclidean Distance-Optimal Post-Processing of Grid-Based Paths

Guru Koushik Senthil Kumar [1], Sandip Aine [2] and Maxim Likhachev [2]
1) Dept. of Mechanical Engineering, Carnegie Mellon University 2) Robotics Institute, Carnegie Mellon University

Carnegie Mellon University

Search-Based Planning Lab

Motivation

Paths planned over grids can often be suboptimal in a Euclidean space and contain a large number of unnecessary turns. In this work, we propose a **novel post-processing technique, called Homotopic Visibility Graph Planning (HVG)** which differentiates itself from existing post-processing methods in that it is **guaranteed to shorten the path to at least as short as the optimal path that lies within the same homotopy class** as the initially computed path.

Method

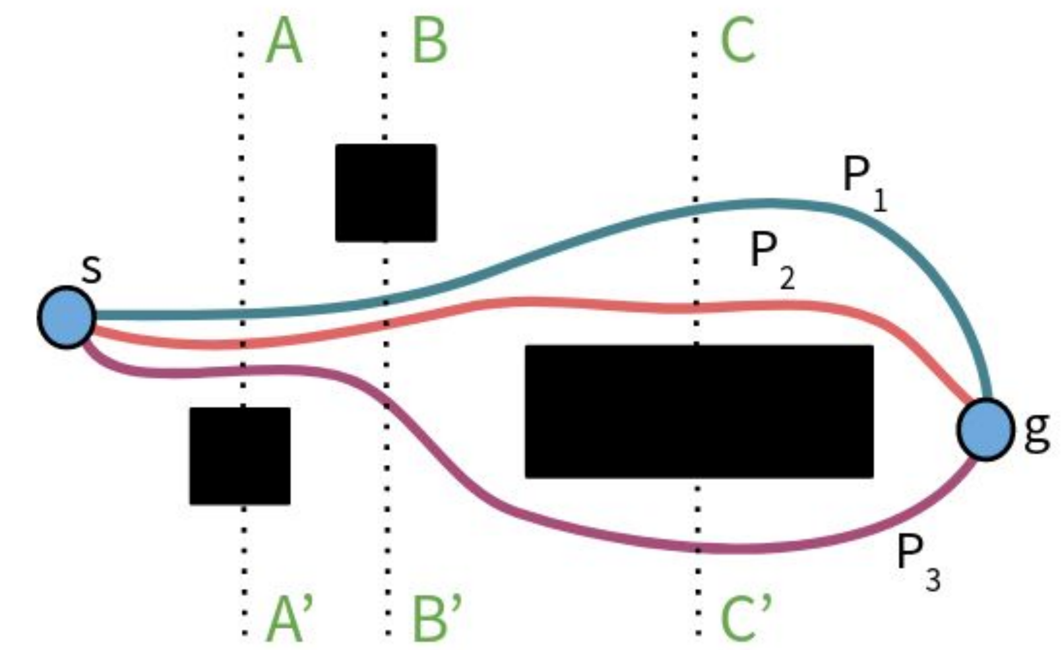


Fig 1: P1 and P2 are in the same homotopy class whereas P3 belongs to a different homotopy class

Core Idea:
Build a **'local' visibility graph** around the given path in its homotopy class. The relevant obstacle corners for HVG have **horizontal and vertical line of sight** to the A* path. Only then does the A* path go **'around' that corner'** and lead to taut paths.

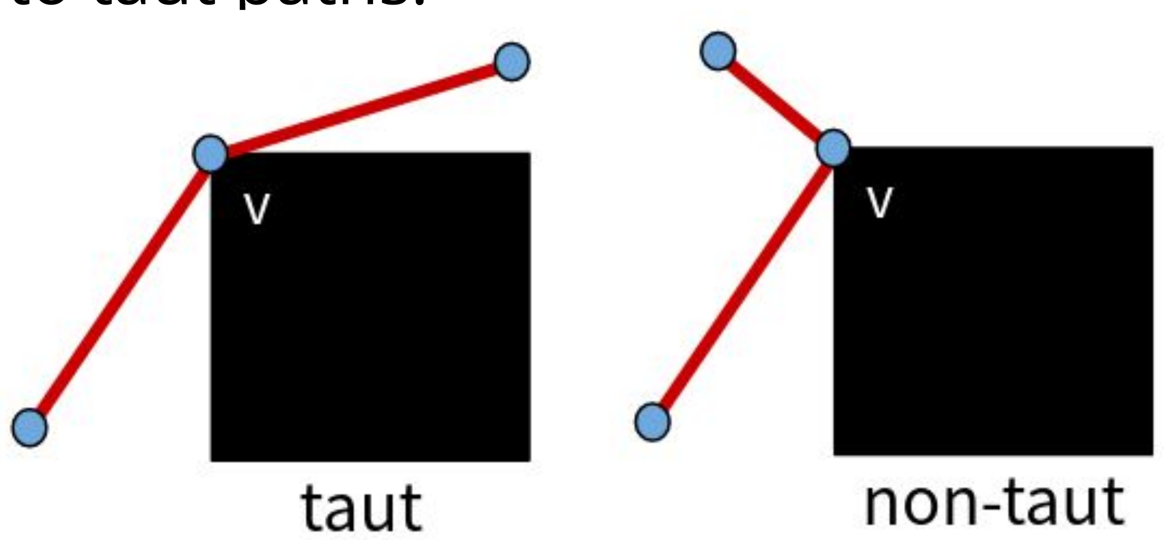
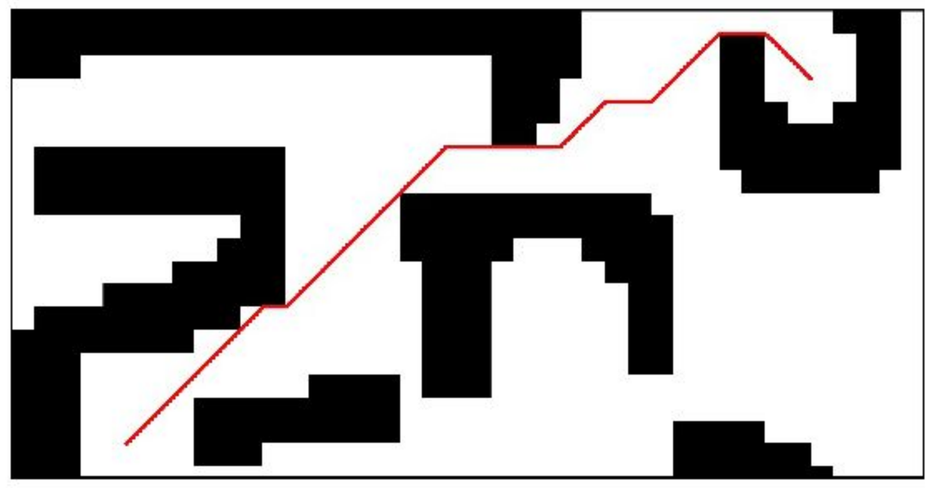
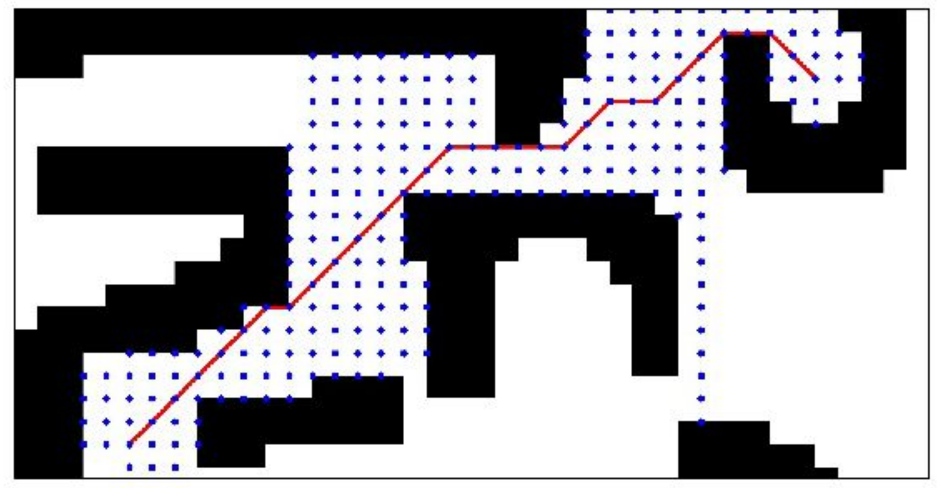


Fig 2: Euclidean optimal paths in 2D always consists of path segments which are taut

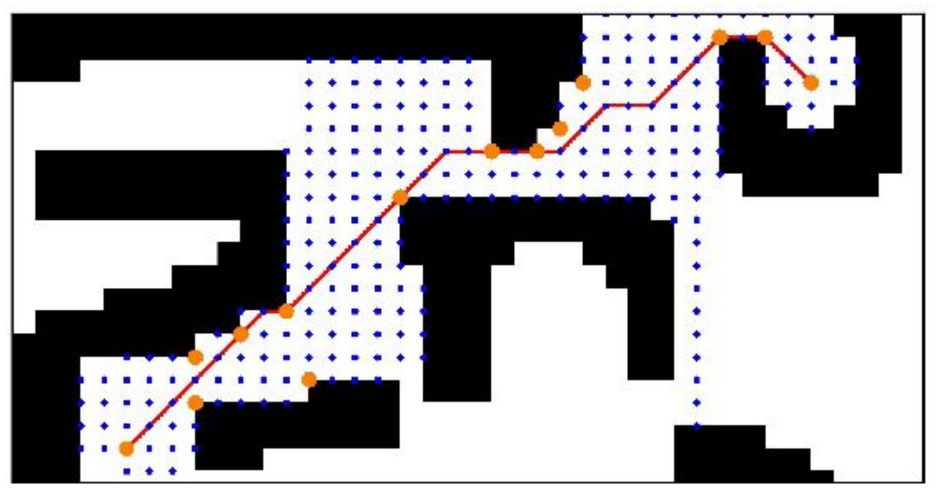
Method



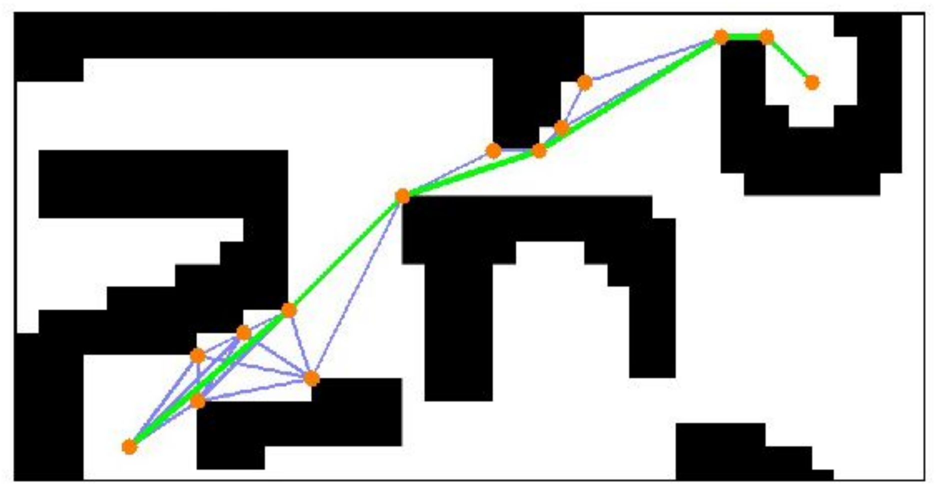
- Obtain a grid-based path using A* or other grid search methods
- It can be seen that this is not Euclidean optimal as there are turns in freespace



- From Lorenzo-Perez (1969), Euclidean optimal paths have vertices at obstacle corners
- From each node in the grid path, scan in all four directions to find obstacle corners



- After scanning, the criteria for choosing HVG vertices is: an obstacle corner which is found in a horizontal and vertical scan from A* nodes



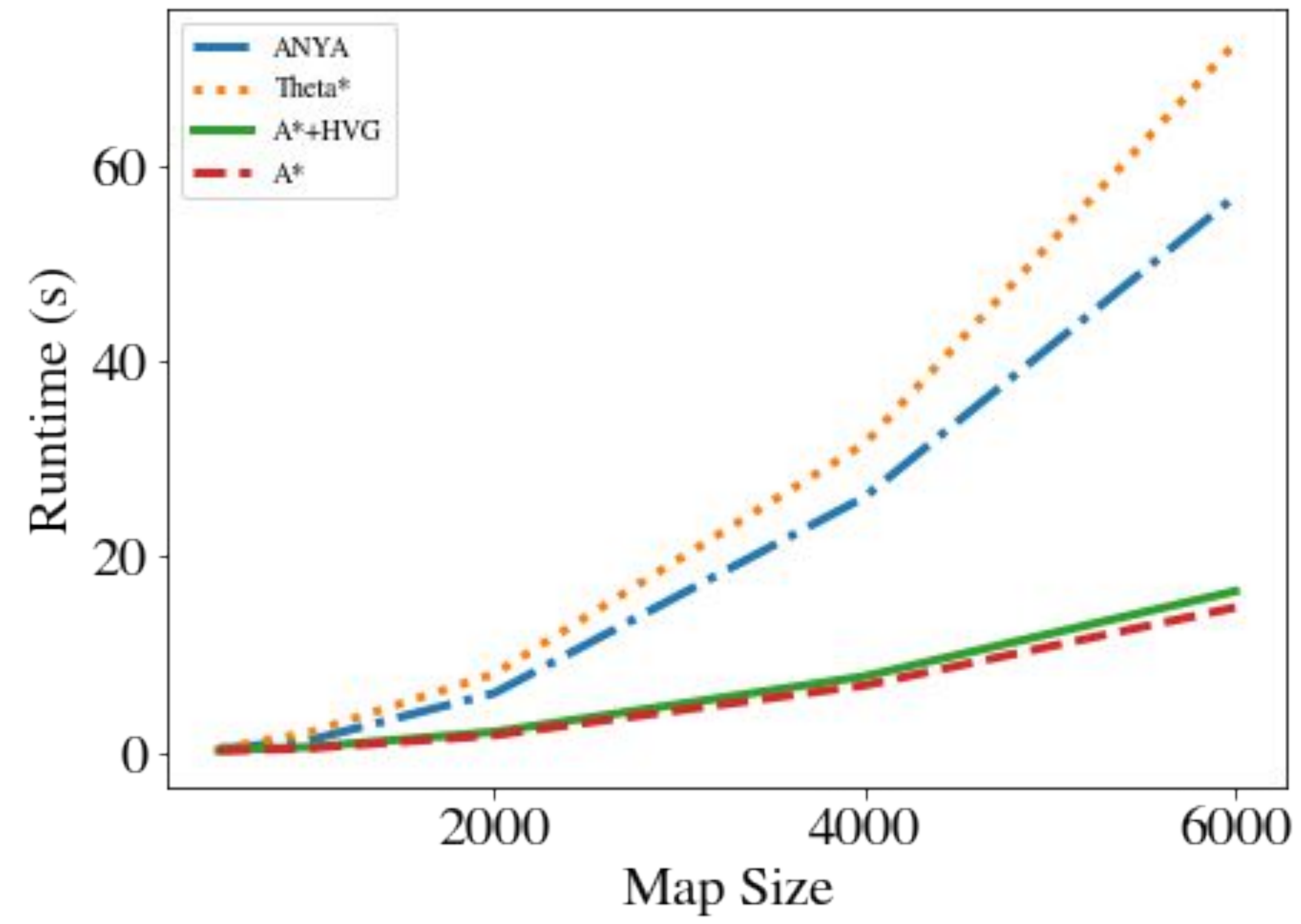
- Build a visibility graph with the HVG corners
- Search the constructed visibility graph to obtain the post-processed path

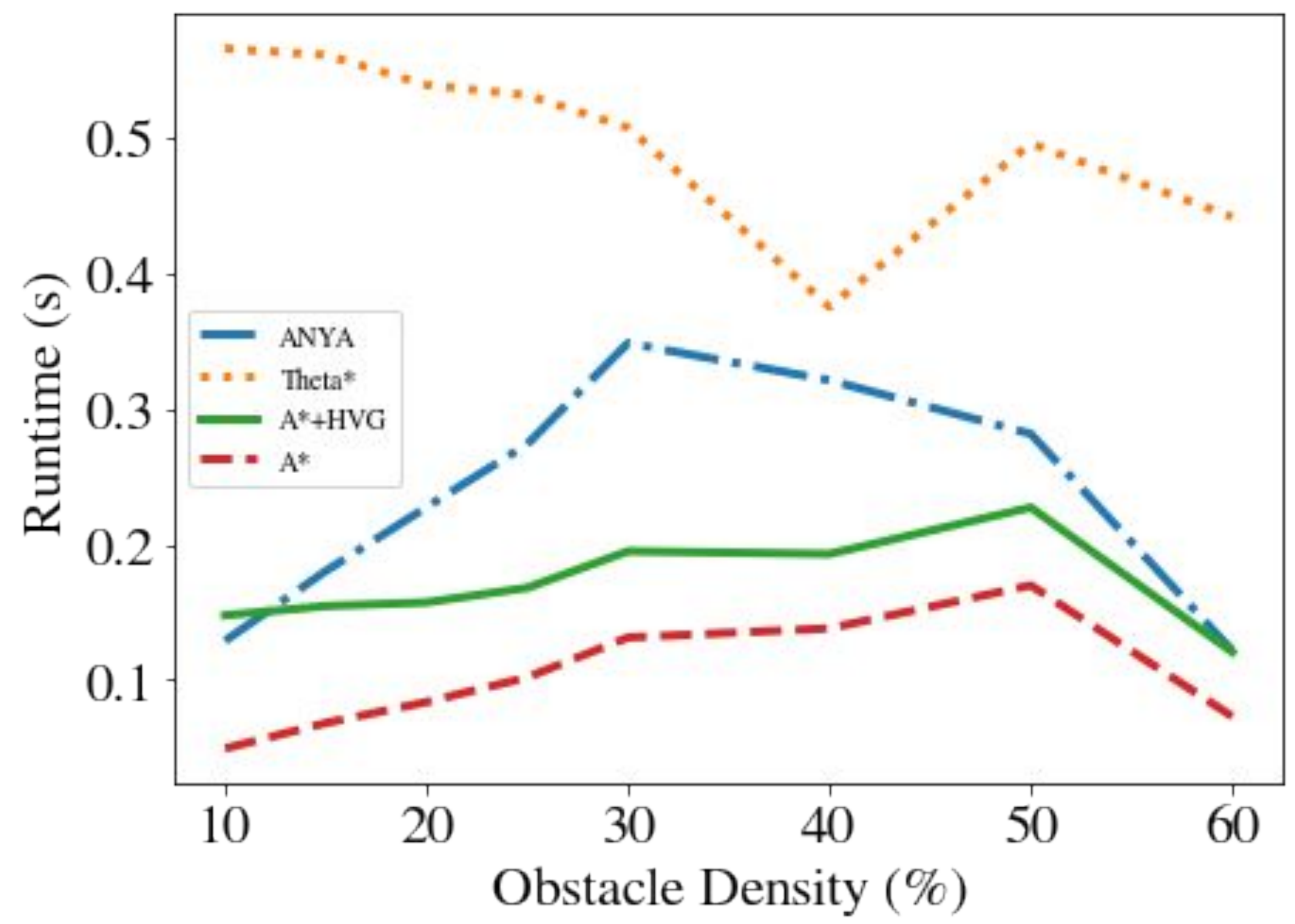
Results

- In maps of size 6000x6000, A*+HVG has **3x faster runtime** than ANYA and Theta*.
- The **path cost** of A*+HVG (optimal in its homotopy class) is **only 0.98% worse** than ANYA (globally optimal) in randomly generated maps of size 6000x6000.
- If weighted A* is used, then **path cost is 9.1% worse** while obtaining a **26x better runtime** than ANYA.
- HVG provides a good tradeoff in large and dense maps.

Plots illustrate the scaling of HVG to large and dense maps compared to post-processing algorithms and any-angle planning algorithms

Results





Conclusions

In this work we developed **a post-processing algorithm** in the grid representation for motion planning which has **theoretical optimality guarantees**. We show empirically that the performance of HVG scales better than any-angle algorithms in **large and dense maps**.